PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-137738

(43)Date of publication of application : 16.05.2000

(51)Int.Cl. G06F 17/30

(21)Application number : 11-140695

(22)Date of filing : 20.05.1999

(71)Applicant : NEC CORP

(72)Inventor : WEN SHAN LEE

(30)Priority

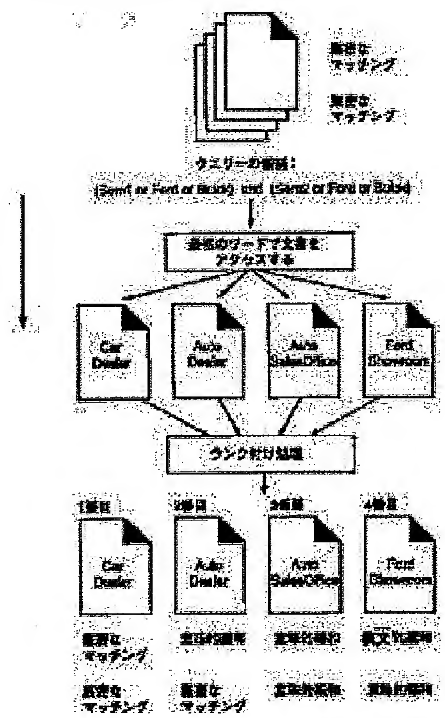
Priority number : 98 185323 Priority date : 03.11.1998 Priority country : US

(54) METHOD AND DEVICE FOR INDEXING PLURAL GRANULARITIES AND SUPPORTING EXPANSION OF QUERY WHILE EFFECTIVELY USING QUERY PROCESSING

(57)Abstract:

PROBLEM TO BE SOLVED: To expand a query not physically but conceptually and to reduce documents related as a result without missing them by performing efficient query processing while using an index in small size, and processing continuous queries.

SOLUTION: The size of index is reduced by merging several entries (tuples) to one entry with the granularity of much higher level. During query processing, that tuple is used for retrieving related documents. Afterwards, the source word of the query is used for ranking documents to be provided as a result during query processing based on severe matching, semantically similar matching and constructively related matching with much higher granularity. Thus, while maintaining entire accuracy in a retrieval mechanism, the size of index can be reduced and more accelerated query processing can be provided.



[Claim(s)]
[Claim 1]A way a word in said index searches a database of a document which is the original degree of fragmentation including a relation between a word included in a preliminary index of a document characterized by comprising the following, and a document, and said index and said word.
A step [/ word / in said preliminary index] transposed more to a concept of a higher rank in order that said method may generate an index of the coarser degree of fragmentation of small size a.
b) A step which extends logically said query applied to a database of a document by [corresponding] transposing to a concept of a higher rank more in a word of a query which has the original degree of fragmentation.
A step which performs said query extended logically and corresponds using said index of the degree of fragmentation coarser than c) and which searches a document relevant to a concept of a higher rank more.

[Claim 2]A search method by which a step which ranks a searched document being further included in claim 1 based on an order of d relevance.
[Claim 3]A search method, wherein a document searched with said rank step is ranked in claim 2 using a word of a query which has the original degree of fragmentation.
[Claim 4]A search method characterized by an order of relevance being the order when not matching when a word of a query and a word included in a searched document make the start a case where it matches strictly, and matches semantically henceforth and it matches syntactically in claim 3.
[Claim 5]A search method characterized more by a concept of a higher rank being a semantic concept of a higher rank more at said replacement step in claim 1.
[Claim 6]A search method characterized more by each of a semantic concept of a higher rank containing a synonym in claim 5.

[Claim 7]A search method, wherein a part of word in a preliminary index which meets the predetermined standard by said replacement step is transposed to a word to which a generic concept corresponds more in claim 1.
[Claim 8]A search method by which being based on whether said predetermined standard has said word in a term dictionary in claim 7.
[Claim 9]A search method characterized more by said concept of a higher rank being a syntactic concept of a higher rank more at said replacement step in claim 1.
[Claim 10]A search method, wherein each of said syntactic concept of a higher rank includes more a word generated within [both] a document in claim 9 exceeding frequency of a certain level.
[Claim 11]A search method having a step replaced with a concept to which a higher rank corresponds more which has a semantic concept of a higher rank only for a word of a query by which said step which extends a query logically meets the further b(i) predetermined standard in claim 1 more.
[Claim 12]Claim 11 comprising:
A step which extends said query still more logically when said step which extends a query logically adds a word which b-(ii)-corresponds, and which is syntactically related more to each of said concept of a higher rank further.
b). A step which extends said query still more logically by adding a word syntactically related to each of a word in a query which is not meeting the (iii) aforementioned predetermined standard

[Claim 13]Claim 12 comprising:
A step by which said step which extends a query logically meets the further a(iv) predetermined standard, which is related in said syntactically related word and which is transposed more to a concept of a higher rank.
a). Said word relevant to a (v) syntax target, and a step removed from a query after extending a portion which becomes redundant among said concepts of a higher rank more

[Claim 14]A search method by which being based on whether said predetermined standard has said word in a term dictionary in claim 13.
[Claim 15]A search method, wherein a word which has two or more meanings in said preliminary index at said replacement step is replaced more by two or more corresponding concepts of a higher rank in claim 1.
[Claim 16]A search method, wherein a word by which said predetermined standard is not met in claim 12 is a proper noun.
[Claim 17]A search method, wherein said execution step is continued in a continuous stage until a corresponding document more relevant to a concept of a higher rank is searched only for a predetermined number in claim 1.
[Claim 18]A search method with which said each stage is characterized by expressing one

extended class in claim 17.

[Claim 19]A search method with which said each stage is characterized by expressing one slot in one extended class in claim 17.

[Claim 20]A search method characterized by searching a document with an order reflecting a level of importance assigned to one word in a query at least in each stage in claim 17.

[Claim 21]How to search a database of a document which includes a relation between a concept of a higher rank, and said index and said concept more corresponding to a word of the degree of fragmentation of origin in which an index of a document characterized by comprising the following with small size and a document are contained

A step to which said method extends logically a query applied to a database of a document by [corresponding] transposing to a concept of a higher rank more in a word of a query of the degree of fragmentation of a origin.

b) A step which performs said query extended logically and corresponds using said index and which searches a document relevant to a concept of a higher rank more.

[Claim 22]A search method having a step replaced with a concept to which a higher rank corresponds more which has a semantic concept of a higher rank only for a word of a query by which said step which extends a query logically meets the further a(i) predetermined standard in claim 21 more.

[Claim 23]A search method characterized more by each of a semantic concept of a higher rank containing a synonym in claim 22.

[Claim 24]A search method characterized more by said concept of a higher rank being a syntactic concept of a higher rank more in claim 21.

[Claim 25]A search method, wherein each of said syntactic concept of a higher rank includes more a word generated within [both] a document in claim 24 exceeding frequency of a certain level.

[Claim 26]Claim 22 comprising:

A step which extends said query still more logically when said step which extends a query logically adds a word which a-(ii)-corresponds, and which is syntactically related more to each of said concept of a higher rank further.

a). A step which extends said query still more logically by adding a word syntactically related to each of a word in a query which is not meeting the (iii) aforementioned predetermined standard

[Claim 27]Claim 26 comprising:

A step by which said step which extends a query logically meets the further a(iv) predetermined standard, which is related in said syntactically related word and which is transposed more to a concept of a higher rank.

a). Said word relevant to a (v) syntax target, and a step removed from a query after extending a portion which becomes redundant among said concepts of a higher rank more

[Claim 28]A search method by which being based on whether said predetermined standard has said word in a term dictionary in claim 27.
[Claim 29]A search method, wherein a word by which said predetermined standard is not met in claim 26 is a proper noun.
[Claim 30]A search method, wherein each of said syntactic concept includes a word generated within [both] a document in claim 26 exceeding frequency of a certain level.
[Claim 31]A search method by which a step which ranks a searched document being further included in claim 21 based on an order of c relevance.
[Claim 32]A search method, wherein said searched document is ranked in claim 31 using a word of a query which has the original degree of fragmentation.
[Claim 33]A search method characterized by an order of relevance being the order when not matching when a word of a query and a word included in a searched document make the start a case where it matches strictly, and matches semantically henceforth and it matches syntactically in claim 32.
[Claim 34]A search method with which a word of the degree of fragmentation of origin contained in a document in claim 21 is characterized by two or more things corresponded more to a concept of a higher rank.
[Claim 35]A search method, wherein said execution step is continued in a continuous stage until a corresponding document more relevant to a concept of a higher rank is searched only for a predetermined number in claim 21.
[Claim 36]A search method, wherein said each stage expresses one extended class in claim 35.
[Claim 37]A search method, wherein said each stage expresses one slot in one extended class in claim 35.
[Claim 38]A search method characterized by searching a document with an order reflecting a level of importance assigned to one word in a query at least in each stage in claim 35.
[Claim 39]A system with which a database of a document whose word in said index is the original degree of fragmentation is searched including a relation between a word included in a preliminary index of a document characterized by comprising the following, and a document, and said index and said word.
An indexer [/ word / in said preliminary index] transposed more to a concept of a higher rank in order that said system may generate an index of size with the small degree of fragmentation coarser than a.
b) A user interface for providing a query applied to a database of said document.
c) A word of a query which has the original degree of fragmentation by [corresponding] transposing to a concept of a higher rank more, A processor which performs said query which extended said query logically and was extended logically using an index of the coarser degree of fragmentation, and corresponds and which searches a document relevant to a concept of a

higher rank more.

[Claim 40]A search system, wherein said processor ranks a searched document in order of relevance in claim 39.
[Claim 41]A search system said processor's using a word of a query which has the original degree of fragmentation, and ranking a searched document in claim 40.
[Claim 42]A search system characterized by an order of relevance being the order when not matching when a word of a query and a word included in a searched document make the start a case where it matches strictly, and matches semantically henceforth and it matches syntactically in claim 41.
[Claim 43]A search system characterized more by said concept of a higher rank being a semantic concept of a higher rank more in claim 39.
[Claim 44]A search system characterized more by each of said semantic concept of a higher rank containing a synonym in claim 43.
[Claim 45]A search system characterized by a thing to which said indexer corresponds only a word in a preliminary index which meets the predetermined standard in claim 39, and which is replaced more with a concept of a higher rank.
[Claim 46]A search system by which being based on whether said predetermined standard has said word in a term dictionary in claim 45.
[Claim 47]A search system characterized more by said concept of a higher rank being a syntactic concept of a higher rank more in claim 39.
[Claim 48]A search system, wherein each of said syntactic concept of a higher rank includes more a word generated in [both] a document exceeding frequency of a certain level in claim 47.
[Claim 49]A search system extending a query logically in claim 39 by [which is a semantic concept of a higher rank more only about a word of a query by which said processor meets the further c(i) predetermined standard / corresponding] transposing to a concept of a higher rank more.
[Claim 50]As opposed to each of said concept of a twist higher rank to which said processor c-(ii)-corresponds further in claim 49, A search system extending said query logically by adding a syntactically related word and adding a syntactically related word to each of a word in a query which is not meeting the c(iii) aforementioned predetermined standard.
[Claim 51]. In claim 50, said processor meets the further c(iv) predetermined standard. By removing said syntactically related word from a query after extending said related word relevant to [transpose to a concept of a higher rank more and] c(v) syntax target and a portion which becomes redundant among said concepts of a higher rank more, A search system extending said query logically.
[Claim 52]A search system by which being based on whether said predetermined standard has said word in a term dictionary in claim 51.

[Claim 53]A search system, wherein a word in said preliminary index which has two or more meanings is transposed more to two or more corresponding concepts of a higher rank in claim 39.

[Claim 54]A search system, wherein a word by which said predetermined standard is not met in claim 50 is a proper noun.

[Claim 55]A search system, wherein execution of said query is continued in a continuous stage until a corresponding document more relevant to a concept of a higher rank is searched only for a predetermined number in claim 39.

[Claim 56]A search system, wherein said each stage expresses one extended class in claim 55.

[Claim 57]A search system, wherein said each stage expresses one slot in one extended class in claim 55.

[Claim 58]A search system characterized by searching a document with an order reflecting a level of importance assigned to one word in a query at least in each stage in claim 55.

[Claim 59]A system with which a database of a document which includes a relation between a concept of a higher rank, and said index and said concept more corresponding to a word of the degree of fragmentation of origin in which an index of a document characterized by comprising the following with small size and a document are contained is searched.

A user interface for said system to provide a query applied to a database of the a aforementioned document.

b) A word of a query which has the original degree of fragmentation by [corresponding] transposing to a concept of a higher rank more, A processor which performs said query which extended said query logically and was extended logically using said index, and corresponds and which searches a document relevant to a concept of a higher rank more.

[Claim 60]A search system extending said query logically by transposing only a word of a query by which said processor meets the further b(i) predetermined standard in claim 59 to a concept which has a semantic concept of a higher rank more, and to which a higher rank corresponds more.

[Claim 61]A search system characterized more by each of a semantic concept of a higher rank containing a synonym in claim 60.

[Claim 62]A search system characterized more by said concept of a higher rank being a syntactic concept of a higher rank more in claim 59.

[Claim 63]A search system, wherein each of said syntactic concept of a higher rank includes more a word generated within [both] a document in claim 62 exceeding frequency of a certain level.

[Claim 64]As opposed to each of said concept of a twist higher rank to which said processor b-(ii)-corresponds further in claim 60, A search system extending said query logically by adding a syntactically related word and adding a syntactically related word to each of a word in a query

which is not meeting the b(iii) aforementioned predetermined standard.

[Claim 65]. In claim 64, said processor meets the further b(iv) predetermined standard. By removing said syntactically related word from a query after extending said related word relevant to [transpose to a concept of a higher rank more and] b(v) syntax target and a portion which becomes redundant among said concepts of a higher rank more, A search system extending said query logically.

[Claim 66]A search system, wherein said predetermined standard is based on whether said word is in a term dictionary in claim 65.

[Claim 67]A search system, wherein a word by which said predetermined standard is not met in claim 64 is a proper noun.

[Claim 68]A search system, wherein each of said syntactic concept includes a word generated within [both] a document in claim 64 exceeding frequency of a certain level.

[Claim 69]A search system, wherein said processor ranks a searched document further in claim 59 based on an order of relevance.

[Claim 70]A search system, wherein said searched document is ranked in claim 69 using a word of a query which has the original degree of fragmentation.

[Claim 71]A search system characterized by an order of relevance being the order when not matching when a word of a query and a word included in a searched document make the start a case where it matches strictly, and matches semantically henceforth and it matches syntactically in claim 70.

[Claim 72]A search system to which a word of the degree of fragmentation of origin contained in a document in claim 59 is characterized by two or more things corresponded more to a concept of a higher rank.

[Claim 73]A search system, wherein execution of said query is continued in a continuous stage until a corresponding document more relevant to a concept of a higher rank is searched only for a predetermined number in claim 59.

[Claim 74]A search system, wherein said each stage expresses one extended class in claim 73.

[Claim 75]A search system, wherein said each stage expresses one slot in one extended class in claim 73.

[Claim 76]A search system characterized by searching a document with an order reflecting a level of importance assigned to one word in a query at least in each stage in claim 73.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

[Detailed Description of the Invention]
[0001]
[Field of the Invention]This invention relates to the field of the index generally applied to collecting the documents in a database, and a query. It is related with reduction of the size of the index used for carrying out effective extension of a query, and processing and extension of a query in more detail, and processing of a continuous query.
[0002]
[Description of the Prior Art]The conventional search system which searches a document is based on the common principle and methodology which classify a document by applying a query. A document is specified a priori by an expert or the librarian, and index attachment is usually manually done using the adjusted term. Index attachment of the document may be carried out again based on the word (word) included in the document. A user connects between them with the word chosen from the term which can be specified with a suitable Boolean operator, and searches a document. A strict matching strategy is used in a such type system. Although this approach has many advantages of being simple and highly precise, the problem of a word mismatch produces it.
[0003]The author is the document, and the problem of the word mismatch in information retrieval is produced by using another word, when a certain word is being used and a user specifies the same concept as it as it in a query, although a certain concept is expressed. Drawing 1 shows that the words used in the document of HyperText Markup Language (HTML) related with "car (passenger car)" and "dealer (store)" may differ among various documents. Languages other than HTML like an extensible markup language (XML) and Standard Generalized Markup Language (SGML) are also used. When a user uses the word of "automobile (car)" and "dealer (store)" for a query, a result with which one cannot search the target document on the problem of a word mismatch is brought.
[0004]In this specification, since the object of search assumes that English is mainly contained, each element of the query used for search is described in English. However, these can also be

expressed in the language of which country according to a user's demand. Here, the meaning in Japanese of the element will be expressed in a parenthesis (accepting necessity) following the element described in said English. Therefore, Japanese in the parenthesis concerned is for only explaining the meaning of the element of a query, and does not affect the result of a query.

[0005]Extension of the query is suggested as a technique which solves such a problem. The word (for example, word for which it has a related meaning of a synonym or others) to which the meaning was [this approach] similar, and a syntactically related word. A query is extended by using (for example, the word group which appears simultaneously in the same document above fixed frequency is a syntactic coincidence word) as a word in a query. In this way, extension of a query will increase a possibility of matching the word in a related document. Use of extension of a query will extend a query including the word of "car dealer (store of a passenger car)" so that the term of the meaning same as follows may be included.

[0006]Line 1. [("car(passenger car)" OR"automobile (car)" OR"auto (car)" OR "sedan (sedan)") OR line 2.] ("Ford(Ford car)" OR "Buick (BYUIKKU vehicle)") AND line 3. ("dealer(store)" OR"Showroom (showroom)" OR "SalesOffice (sales store)").

[0007]There are two types of extension of the query contained in the above-mentioned example. Extension of the query of the line 1 and the line 3 adds the additional word relevant to "car" and "dealer" in a definition. That is, a semantically similar word is added. "automobile", "auto", and "sedan" are words which has a meaning similar to the word of "car." Similarly, "Showroom" and "SalesOffice" are words which has a meaning similar to the word of "dealer." Extension of the query of other types is shown in the line 2.

This is based on syntactic cooccurrence relation.

Many words used with World Wide Web (it is also only called a web) are proper nouns actually.

It is not found in a term dictionary.

For example, a proper noun is called Ford, Buick, NBA, and NFL (National Football League). As mentioned above, syntactic cooccurrence relation is drawn by analyzing the frequency where two words appears simultaneously in the same document. This is based on assumption that a possibility that those words are related is high, when two words appears in the same document frequently. as the word generated with "Ford" -- "dealer (store)" "body shop (a body factory", "Mustang (Mustang: name of the car by Ford Co.)", "Escort (escort: name of the car by Ford Co.)", etc. can be considered.)

[0008]In order to support extension of a query, the index of the word associated by the definition and a syntactic relation like coincidence information must be maintained appropriately. The index related with a word by the definition is constituted as a hierarchy cluster of a layered structure, a semantic network, or a related word. . About said layered structure, were carried out in Athens, Greece, in August, 1997. the 23rd International. Conference on Very Large. The page 538-547 of the proceedings of Data Bases, and W.

Please refer to "Facilitating Multimedia Database Exploration through Visual Interfaces and Perpetual Query Reformulations" besides Li. About said semantic network, 1990 and International Journal of Lexicography 3 (4), G. A. Miller in the page 245-264 "Nouns in WordNet : Refer to A Lexical Inheritance System." About the hierarchy cluster of a related word. Refer to "The SMART and SIRE Experimental Retrieval Systems" by G. Salton of New York, McGraw-Hill, and the page 118-155, etc. in 1983. Since a syntactic relation like syntactic cooccurrence relation is expressed with binary relations, the size of a syntactic-related index is dramatically large. Some techniques are proposed in order to solve this problem. The proceedings of the Fifteenth annual International ACM SIGIR Conference [in / techniques / such / 1992 and Denmark], G. "Use of syntactic context to produce term association lists for text retrieval" by Grefenstette, The proceedings of the 19 th Annual International ACM SIGIRConference in Zurich in 1996 and Switzerland, J. "Query Expantion Using Local and Global Document Analysis" by Xu etc., . It can set to Philadelphia, American Pennsylvania, in 1997. Refer to "Guessing Morphology from Terms and Corpora" by C. Jacquemin of the proceedings of the20 th Annual International ACM SIGIR Conference. Such a technique includes analysis of occurrence frequency, and use of a morphological rule (for example, all the words are changed into the gestalt used as the origin), or a term dictionary.

[0009]About the problem of a word mismatch, remarkable research has been done in the field of information retrieval (IR). About this, 1983 and McGraw-Hill BookCompany issue, G. "Introduction to Modern Information Retrieval" by Salton etc., 1989, Addison-Wesley Publishing Company, and Inc issue, G. It is based on Salton. "Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer", and 1997, Refer to "Readings in Information Retrieval" by K. Sparck Jones of San Francisco, American California, and Morgan Kaufmann, etc.

[0010]However, these the researches of most point to points about the standard of search, such as precision and recall. How to support extension of a query effectively (in 1993, the proceedings of the 3 rd Text Retrieval Conference in State Gaithersburg of Maryland) C. Although there are some researches which suggested the mechanism of refer to "Automatic Query Expansion Using SMART" or index attachment by Buckley etc., two problems without the solution to satisfy still remain. The 1st problem is a proper noun with many separate words in a set (for example, web) of a certain document.

Since many each words have the same word and the syntactically related word semantically, the size of an index is becoming very large.

Since a query is extended by the additional word, the 2nd problem is that the cleanup cost of a query becomes high.

[0011]Since the number of documents increases dramatically, and the word currently used is very various, and it is inconsistent and is occasionally wrong when dealing with the document information collected from the web (for example, type error), these problems become increasingly remarkable. In a certain research, almost all the user query about a web usually

has two words. About this, they are proceedings of 1995 and Digital Libraries (DL'95), B. Croft etc. "Providing Government Information on the Internet : Please refer to Experienceswith THOMAS." However, if query extension is used, the length of a query will become long substantially. As a result, most existing search engines on a web can provide query expanded function.

[0012]Here, the existing research in the field of query extension is outlined. Query extension attracted remarkable attention in the field of IR. However, the portion which has attracted attention until now was evaluating the grade of the standard (namely, precision and recall) of the search improved by extension of a query. Another research has focused on building a dictionary, in order to identify 1 set of similar terms about the word of the given query. However, old research is not tackling the point of making small the problem of efficient processing of a query when a query is extended, and size of the index used for performing extension and processing of a query. The problem which ranks a document based on strict matching and resemblance matching is left behind as a difficult thing.

[0013]SMART is one of the advanced information retrieval systems known well. About this, 1971, American New Jersey Englewood. The SMART Retrieval System -Experiments in Automatic Document Processing of the Gerard Salton edit published from Prentice-Hall of Cliffs, "Experiments with a fast algorithm for automatic classification" by R. T. Dattola of Chapter 12, And refer to "The SMART and SIRE Experimental Retrieval Systems" by G. Salton of the above-mentioned literature, etc. Each document is expressed in SMART by the terminological vector. Each position of the vector expresses the dignity (importance) of the corresponding term in a document. a set of the document of M individual which has N different terms is expressed with the procession of MxN. A query is also expressed as a terminological vector. Search of a document is due to calculation of the similarity corresponding to the cosine of query vectors and the vector of each document. INQUERY is among the systems known well [other]. About this, it is 3:327-332 of 1995 and Information Processing and Management, J. Refer to "Trec and tipster experiments with inquery" by Callan etc.

[0014]A potential meaning index (LSI) is the technique depending on the conceptual index by matching like a dictionary statistically drawn instead of individual term search. About this, 1990, Journal of the America Society of Information Science, "Indexing by latent semantic analysis" by R. Harshman of 41:391-407, etc., By and the proceedings of 1995 and the 1995 ACM Conference on Supercomputing. M. Please refer to "Computational Method for Intelligent Information Access" by W. Berry etc. It assumes that LSI has some structures which are not in sight, i.e., a potential structure, in directions of a word, and the structure needs to be exteriorized by analyzing generating of the word in a document. Therefore, a document is considered as a vector in the term space of the very big range, and each element of the vector expresses the occurrence frequency of the specific term in the given document. The standard based on the whole and local weighting refined more is also used, and it gets. The shortened singular value decomposition (SVD) evaluates the structure of word use covering a document. Please

refer to the 2nd edition of "Matrix Computations" by G. Golub of Johns-Hopkins of American Maryland state Baltimore, etc. for this in 1989. Here, a search is performed using the database which has a singular value, and the vector acquired from shortened SVD. Let approach of this information retrieval be a standard coarser than the thing based on each term in the preliminary evaluation of LSI.

[0015]The automated query extension has been suggested for a long time as a technique which deals with a word mismatch problem. They are proceedings of the 17 th Annual International ACM SIGIR Conference performed about this in Republic of Ireland Dublin in 1994, E. Please refer to "Query Expansion Using Lexical-Semantic Relations" by Voorhees. In a certain approach, a possibility that a query is extended using a thesaurus and a word matches within a related document is improved. In research, it turns out that an improvement has a limit only by using an only common thesaurus. Much innovative technique is also proposed. The proceedings of 1994 and the 3 rd International ConferenceonInformation and Knowledge Management, O. "Query Expansion Using Domain Adapted, Weighted Thesaurus in an Extended Boolean Model" by Kwon etc., The proceedings of the 16 th Annual International ACM SIGIR Conference performed in Pittsburgh, American Pennsylvania, in 1993, E. "Concept Based Query Expansion" by Voorhees, "Query Expansion Using Lexical-Semantic Relations" by E. Voorhees of the proceedings, And please refer to "Computational Methods for Intelligent Information Access" by M. W. Berry of the proceedings, etc. By the automated query extension, the increase in efficiency of 25% of search is calculated from 7% on the average as a result of the experiment. Please refer to "Automatic Query Expansion Using SMART" by C. Buckley of the proceedings, etc. for this.

[0016]Improvement of a query is attained also by including a syntactically related word. This approach carries out clustering of the word based on the coincidence information within a document, and extends a query using these clusters. Since this coincidence information is binary relations, the size of such an index will become always very big. A certain group used the compilation of the coincidence statistics about modification of a word, and changed or generated SUTEMA (stemmer), and it was proved [group] which is advantageous compared with the approach only using a morphological rule. Please refer to "Corpus-Specific stemming Using Word Form Co-occurrence" besides W. B. Croft of the proceedings of the Fourth Annual Symposium for this in 1994. Each above-mentioned technique which extends the term of a query to 1 set of semantically related terms is called whole (global) analysis. In query extension, the term from relevance feedback is also added to a query, and the efficiency of search is improved. June, 1990, Journal of the American. Refer to "Improving retrieval performance by relevance feedback" besides 41(4):288-297 of Society for Information Science, and G. Salton. This is called partial (local) analysis. By old research, by applying the whole analysis technique which used the context of a word and the structure of words and phrases to some groups of a document shows that search results more effective than simple local feedback and more positive are obtained. For details, refer to "Query Expansion Using

Local and Global Document Analysis" by J. Xu of the above-mentioned literature, etc.
[0017]However, as mentioned above, it does not aim at that old research makes small size of the index used for solving the problem of efficient processing of a query when a query is extended, or performing query extension and query processing.

[0018]

[Problem(s) to be Solved by the Invention]The purpose of this invention is to provide the method and device which perform efficient query extension using the index of small size, and process a continuous query, in order to solve the problem of a word mismatch, and inefficiency [of the query processing produced as a result]. It is semantically similar in more detail with the word specified in the query, the query is extended physically and notionally using the word which has relation syntactically, and it lessens missing a document related as a result.

[0019]In order to support extension of a query, the index of the word related about a definition and the word in syntactic cooccurrence relation needs to be maintained, and the following two problems become important about support of such query extension. The 1st is a problem of the size of an index table and the 2nd is a problem of the overhead of query processing. This invention also makes it the purpose to solve these problems.

[0020]

[Means for Solving the Problem]According to this invention, a concept and treatment structure of information which consist of two or more degrees of fragmentation are used in order to support extension of a query. This invention contains an index attachment phase, a query processing phase, and a rank phase. In an index attachment phase, grouping of the semantically similar word is carried out as one concept, and one actual index size becomes small as a result for a semantic concept subdivided in this way more coarsely. A word between query processings and in a query uses the contents of a dictionary and actual data, it is mapped by a corresponding semantic concept and syntactic extension, and logical extension is performed to the original query as a result. An overhead about processing is avoided. Next, a word of the first query is used for ranking a document obtained as search results based on strict matching, semantic matching, and syntactic matching, and is used also for performing processing of a continuous query.

[0021]

[Embodiment of the Invention]The method for extending a query efficiently and the suitable embodiment of a device which are depended on this invention are described in detail below with an accompanying drawing. Although the following explanation is made about the PERICO object oriented database managerial system (OODBMS) of NEC, it should be cautious of this invention not being what is restricted to this. This invention is applied to the aggregate of various database systems and a document, and it deals in it.

[0022]This invention provides effective index attachment and processing support about extension of a query by introducing the concept of two or more degrees of fragmentation. The approach of this invention sets up an index about the word which is semantically similar after

stemming (stemming) of a word using an available technique, and a syntactically related word. About said technique, the proceedings of the 19 th AnnualInternational ACM SIGIR Conference in 1996, Switzerland, and Zurich, J. "Query ExpansionUsing Local and Global Document Analysis" besides Xu, And the proceedings of the 20 th Annual International ACM SIGIR Conference in 1997 and Philadelphia, American Pennsylvania, C. Refer to "Guessing Morphology from Terms and Corpora" of Jacquemin. The approach of this invention makes size of an index small by being the degree of fragmentation of a high level more, and merging some entries (tuple) into one entry. The tuple with the information on the degree of fragmentation of a higher level is used for searching a related document between query processings. Then, the original word of a query is the finer degree of fragmentation, and since the document obtained as a result between query processings based on strict matching, semantically similar matching, and syntactically related matching is ranked, it is used. Maintaining the accuracy of the whole in a search mechanism by using the index and query processing technique which have two or more degrees of fragmentation, size of an index can be made small and quicker query processing can be realized.

[0023]It explains that first it is adapted with the notation of two or more degrees of fragmentation in relation to the conventional index attachment currently used by almost all IR system how. Next, the estimate about the overhead over the memory location in the case of performing index attachment which has two or more degrees of fragmentation about a set of a predetermined document is performed.

[0024]In order to search a given word easily from a document list, the conventional IR system holds an index and extracts the group of the word simultaneously related with the obtained document. In this case, the term of a "document" should be cautious of relating to the combination of a text, an image, or a text and an image.

[0025]Drawing 2 shows the example of the index. The table shown in (b) of drawing 2 is the index which transposed the table shown in (a) of drawing 2. At drawing 2, in order to explain easily, these indexes are shown in the form of the table. However, in a actual environment, the class of the upper layer of PERCIO OODBMS of NEC is used, for example. If the example of one query is taken, a user will use a word "car (passenger car)" and "dealer (store)" first, and if a query is created, IR system will take out a document list from the line to which the table of (b) of drawing 2 corresponds. In this case, the answer of a query serves as an intersection of the document list obtained from two lines. The approach against this IR is what supports only strict matching clearly, A related document including the term which has similar meanings, such as "automobile dealer (store of a car)", "car showroom (showroom of a passenger car)", or "automobile showroom (showroom of a car)", cannot be obtained. Query extension is used from the description [query] "car" and "dealer" in relation to the special utility extended to the description ("car" or "automobile") and ("dealer" or "showroom"). Although this approach is realizable, a remarkable overhead will be invited to query processing. The lookup of several times is needed about each of a word similar as especially semantically instead of 2 times of

the lookups about the index table of (b) of drawing 2 as the word in the original query. A thesaurus tool like an on-line dictionary is required to extend the term of a query to them and a semantically similar term. From these observation, when this invention searches a set of a document, it provides the more effective method of supporting extension of a query.

[0026]As stated previously, in order to avoid the mismatch of user's vocabulary and the author's vocabulary, the extension of a query based on the method of extending a query using the word to which a meaning is similar, and the word which has syntactic relations is needed.

[0027]Drawing 3 shows the data structure for which an addition is needed making extension of a query easy in the conventional IR system. Especially drawing 3 shows the table drawn from the on-line dictionary by which grouping is carried out to the concept to which each word is semantically similar, and including a definition. The table shown in drawing 3 is simplified for explanation. For example, the group "car (passenger car)" of a similar term, "auto (car)", "automobile (car)", and "sedan (sedan)" are expressed as one symbolic entity and sem1. Unlike the semantic resemblance based on a dictionary or a thesaurus, the syntactic relation to IR is determined by the collection of a document itself. Especially the coincidence information on a word is used for associating two words syntactically. Drawing 3 (b) has illustrated the index showing this information. With the auxiliary index of drawing 3, a fundamental query extension technique is supported in IR system by using the conventional IR index of drawing 2. Fundamentally, if a user's query is given, the word list of a query will be extended so that a semantically similar word and a syntactically related word may be included.

[0028]Although an above-mentioned method is used for processing of the query using extension of the query, in this approach, the overhead about processing will become high. According to this invention, the index structure of the addition which can process a query more efficiently is used. The fundamental way of thinking of the approach of this invention changes the index of drawing 2 and drawing 3 so that a query may be extended notionally. Namely, the list of the word of a query is not physically extended by including a semantically similar word and a syntactically related word in a list, A query is notionally extended for the word of a query by [the / related] changing for the semantic concept of an upper level, and a syntactic-related (for example, cooccurrence relation) word more. This brings about addition of the capacity overhead by an additional index structure. However, since a user's query is processed more efficiently, saving can be attained as the whole.

[0029]As mentioned above, in order to process the extended query, an index table is changed as shown in drawing 4. Especially the index table shown in (a) of drawing 4 is drawn from (a) of drawing 2 by transposing each word (it is not a peculiar name) to the word of the semantic concept of an upper level more. The index table shown in (b) of drawing 4 is obtained in the word shown in (b) of drawing 2 by [to which they correspond] combining with the word of the semantic concept of an upper level more, and merging the entry of each document list. Therefore, the line entry corresponding to "car", "auto", "automobile", and "sedan" is expressed with (b) of drawing 4 as single entry Sem1. Similarly, the line corresponding to "dealer",

"showroom", and "SalesOffice" of drawing 2 (b) is summarized to one line of a label called Sem2.

[0030]The index to a syntactically related word is usually quite larger than the index to a word semantically related from several reasons. Many words on a web are peculiar names, and is not found in a dictionary. In the experiment, when the document of 2,904 was analyzed, 42% of keywords were found by WordNet. WordNet is an on-line dictionary which has 60,000 or more words. About this, 1990 and International Journal of Lexicography 3 (4), It is based on G. A. Miller of the page 245-264. "Nouns in WordNet : Please refer to A Lexical Inheritance System." 58% of word of the remainder includes the proper noun and the type error, and this has become the origin which hypertrophies the size of an index. In the conventional IR system, syntactic correlation is usually grasped by cooccurrence relation. Since the cooccurrence relation of the word within the same document is 1 to 1 relation, when n words is identified, the size of an index is set to (nx (n-1)) / 2 in the worst case. Carrying out index attachment of the cooccurrence relation of three or more words for the overhead of a huge memory location and index attachment requires cost dramatically.

[0031]The word (that which is semantically meaningful) found in the dictionary is set to S, and other words (proper noun) of all the are set to P. The cooccurrence relation between words is classified into three different categories based on the above-mentioned classification of the word in a dictionary, and the word which is not in a dictionary.

[0032]- P-P type: (name of Toyota cars), for example, (Toyota (Toyota), Avalon) (Acura (Aqura), Legend (name of the Aqura vehicle)) (Nissan (Nissan), Maxima (name of Nissan cars)),

[0033]- S-P type or P-S type:, for example, (Buick (name of Ford cars), car (passenger car)), Buick, dealer (store), (car, Ford) (Ford, auto (car)) (Ford, dealer) (Ford Co.),

[0034]- S-S type:, for example, (car, garage (garage)) (auto, garage),

[0035]Usually, it is difficult to change the P-P type entry which is not convertible for the coarser degree of fragmentation shown in (b) of drawing 3. However, other entries of all the have S word which can be replaced by the corresponding semantic concept of a higher level. The size of a coincidence index decreases and speedup of query processing is realized by this. Reduction of the size of an index is produced as follows. To each S-P type (w_i, X) entry, all the entry of (w_i, X) shown in (b) of drawing 3 is replaced by (Sem_i, X) of (c) so that w_i may correspond to semantic concept Sem_i. [of drawing 4] Here, the list of corresponding documents is also merged. The same procedure is applied also to a P-S type entry. As shown in (c) of drawing 4, an entry (Ford, car), and (Ford, auto) are replaced by (Ford, Sem1). Similarly, an entry (Ford, dealer), and (Ford, showroom) are replaced by (Ford, Sem2). Such a merge mechanism is explained using (a) of drawing 5, and (b).

[0036]A S-S type entry is merged by the following two methods.

[0037]- Single merge : merge of the type of many [one pair] / many pairs 1 as shown in (a) of

drawing 5, and (b). For example, an entry (car, dealer), (automobile, dealer), and (auto, dealer) are replaced by (Sem1, dealer). The algorithm used here is the same as what is used with a S-P type and a P-S type.

[0038]- Compound merge : merge of a many to many type as shown in (c) of drawing 5. For example, an entry (car, dealer), (automobile, showroom), and (auto, SalesOffice) are replaced by (Sem1, Sem2). The algorithm of this type of merge is as follows.

[0039]1. -- to each S-S type entry (w_i, X), all the entries [of drawing 3] of (b) of (w_i, X) are shown in (c) of drawing 4 so that w_i may correspond to semantic concept Sem_i -- as (Sem_i, X) -- it replaces.

[0040]To each entry of the type of 2. (Sem_i, w_j), w_j replaces (Sem_i, w_j) of such all by (Sem_i, Sem_j) so that it may correspond to semantic concept Sem_j.

[0041]The above-mentioned step 2 should be cautious of the ability to also perform before the above-mentioned step 1. It is carried out repeatedly and deals in Step 1 and Step 2 of this algorithm until what is merged is lost.

[0042]If two or more entries are merged, the syntactic word list of each entry will also be merged by the merger (UNION) operation according to it.

[0043]Index attachment technique which has two or more degrees of fragmentation is mounted in the upper layer of OODBMS, and it deals in it. In such mounting, the table shown in (a) of drawing 2, (a) of drawing 3, and (c) of drawing 4 is a class which has the contents. Other tables are classes which have only a pointer. Updating to an index, deletion, and inserting operation are performed by OODBMS via the program which transmits between automatic-checking-and-continuous-monitoring maintenance or a class. Maintenance of an index which has two or more degrees of fragmentation is performed cumulatively, and reorganization is not needed.

[0044]Next, the estimate of the example by this invention is calculated besides the index based on the conventional word, taking into consideration the overhead of a memory location added since it is required to support the index table based on a semantic concept. As mentioned above, the table shown in drawing 4 is introduced for efficient query processing. First, calculation about the estimate of the memory location about the index used by the conventional IR system, i.e., the table shown in drawing 2, is performed. The number of the documents in a predetermined aggregate presupposes that it is D. The numbers of words (number after removing a stop word and a grouping word using word stemming) in a dictionary in the aggregate of the predetermined document are W, and set to V the number of the words which is not in a dictionary. The average of the numbers of words in the dictionary for every document is set to w, the average of the numbers of words which are not in a dictionary is set to v, and the average of the document number for every word is set to d. The size of an index is calculated based on the number of entries (namely, the number of lines), and the size (namely, the number of pointers) of the whole table. Each entry of the table should be cautious

of being expressed as pointer data. When these parameters are given, the size of the table shown in (a) of drawing 2 is expressed with the following formulas (2).

[0045]

The number of lines [2 (a)] =D ... (1)

Whole size [2 (a)] =(1+v+w) D ... (2).

[0046]Although the word where one pointer is needed for discernment of a document and which is not in a dictionary in the list of a word in each line is expressed, Although v pointers are needed on an average and also the word which is in a dictionary in the list of a word is expressed, since w pointers are needed on an average, it should be cautious of the paragraph of (1+v+w) having arisen. Similarly, the size of the table shown in (b) of drawing 2 is expressed with the following formulas (4).

[0047]

The number of lines [2 (b)] =W+V ... (3)

Whole size [2 (b)] = (1+d) and (W+V) ... (4).

[0048]Each line of this table is an average and needs d pointers which serve as an identifier of a document within a document list, and one pointer which points out the word itself.

[0049]Next, the memory location overhead of an on-line dictionary and a syntactic coincidence table required to support fundamental query extension is estimated. It is considered as the compression element obtained by carrying out grouping of the word which is in a dictionary about f to a semantic concept. Therefore, f can be regarded as a number of a word of averages by which grouping was carried out to one concept. The size of the table shown in (a) of drawing 3 can be expressed like the following formulas (6).

[0050]

The number of lines [3 (a)] =W/f ... (5)

Whole size [3 (a)] =W+W/f ... (6).

[0051]Since the memory location of the word in a dictionary is compressed based on the compression element f, a formula (5) is expressed in this way. It is shown that a formula (6) needs W pointers to express the word in the list of a word, and it is required for the pointer of a W/f individual to express a semantic identifier. The size of the table shown by (b) of drawing 3 is expressed with the following formulas (8) when the worst.

[0052]

The number of lines [3 (b)] =V(V-1) /2+VW+W(W-1) /2 ... (7)

Whole size [3 (b)] =(1+2+q) - (V(V-1) /2+VW+W(W-1) /2) ... (8).

[0053]In a formula (7), the 1st paragraph corresponds to the cooccurrence relation of a P-P type word, the 2nd paragraph corresponds to a S-P type or a P-S type, and the last paragraph corresponds to S-S type cooccurrence relation. q expresses the mean number of the entry in a document list for every paragraph showing cooccurrence relation. Since a syntactic term identifier is expressed, three pointers are needed, and two words is included in the cooccurrence relation of each line.

[0054]Next, it estimates about the memory location overhead about index attachment which has two or more degrees of fragmentation based on this invention which carries out grouping of the group of a semantically similar term to one unique semantic concept. As mentioned above, in order to calculate the size of the index table shown in drawing 4, it is necessary to estimate the average document number for every semantic concept, and the mean number of the semantic concept for every document. It can be shown that the mean number of the document for every semantic concept becomes larger than d, and this extension does not become f-d since two or more terms are cut down by one semantic concept. On the other hand, the mean number of the concept for every document does not exceed w. It can be shown that this number actually becomes a thing similar to w. Based on these parameters, the memory location overhead of an addition of index attachment which has two or more degrees of fragmentation is calculable. The calculation about the table shown in (a) of drawing 4 is as follows.

[0055]

The number of lines [4 (a)] =D ... (9)

Whole size [4 (a)] =(1+v+w) D ... (10).

[0056]That is, size is the same as the table shown in (a) of drawing 2. On the other hand, the size of the table shown in (b) of drawing 4 is as follows.

[0057]

The number of lines [4 (b)] =W/f ... (11)

Whole size [4 (b)] = (1+df) and W/f ... (12).

[0058]Since a word is unified by the semantic concept, the number of the entries of the word in a dictionary decreases based on the element f. However, only the almost same part as the element increases the number of the documents for every semantic concept. As a result, the size of this table becomes the same thing as the table shown in (b) of drawing 2. In the degree of fragmentation of a higher level, the table indicated to be (a) of drawing 4 to (b) should be cautious of it being a table indicated to be (a) of drawing 2 to (b), respectively. Finally, the estimate of the memory location of the table shown in (c) of drawing 4 is calculated as shown in the following formulas (13) and (14).

[0059]

The number of lines [3 (b)] =V(V-1) /2+V, and(W/f)+ (W(W-1) /2f^2) ... (13)

Whole size [3 (b)] =(1+2+q) -V(V-1) /2+ (1+2+qf)

- Vand(W/f)+ (1+2+qf) (W(W-1) /2f^2) ... (14)

.

[0060]Fundamentally, S-S types, S-P types, or all the P-S type coincidence terms are compressed based on the element f, and it becomes small capacity substantially compared with the table shown in (b) of drawing 3.

[0061]Eventually, according to this invention, it is needed except the table shown in (b) of drawing 3. On the other hand, a fundamental query extension technique needs all the tables

shown in drawing 2 and drawing 3. Therefore, although only the increment of the memory location of the table showing the cost about the memory location at the time of adopting the method of this invention in (b) with (a) of drawing 4 becomes large, since the size of the table shown in (c) of drawing 4 becomes small, it compensates for a part for said increase in cost selectively. It depends for the exact numerical value of saving on the value of various parameters mentioned above. Even when the worst, an additional memory location is quite smaller than the twice of the memory location at the time of using a fundamental query extension technique.

[0062]The above-mentioned index attachment technique assumes having only a meaning with a single word, and it argues about it.However, a word usually has two or more meanings. For example, the word of "bank" is interpreted as a financial institution (bank) or a riverside. In order to take into consideration about the word which has two or more meanings, the word (shown by drawing 3) of a semantic word list shall belong to two or more conceptual numbers shown in (a) of drawing 4. For example, "bank" shall be related with Sem10 and Sem20. In order to perform extension of a query in consideration of two or more such meanings, when a query includes one word belonging to several different conceptual numbers, each of the different conceptual number should be taken into consideration in the case of processing of a query.

[0063]In the above-mentioned explanation, index technique is mounted in the upper layer of OODBMS of NEC, and the word in a semantic word list is related with the conceptual number by the pointer. Redundant data is not memorized but its cost of the memory location about a pointer is very low. WordNet provides the synonym to a certain word by the interpretation of various meanings, and ranks it according to the frequency where the meaning is used. For example, more "bank(s)" is interpreted as a financial institution rather than a riverside. The most general semantic interpretation is used for the present execution. However, a data structure is also extensible.

[0064]The grouping of a meaning except having been stated above can be taken into consideration. Only extension of the query by a synonym is taken into consideration in drawing 4. Relaxation of meanings of other molds, such as ISA and IS_PART_OF, can also be taken into consideration. Two or more tables of the gestalt shown in (a) of drawing 4 are generable about the grouping (one related with IS_PART_OF one about ISA for example) of various meanings. One table can also be used about the grouping of various meanings. When extending a query by both the synonym and a hypernym, a lookup is performed to two or more tables.

[0065]In order to cope with the problem of a word mismatch, the query processing technique needs to extend the word of a query using a related word. As a result, by the relevance over the word of the original query, the additional task which ranks a document is performed and it gets. Next, processing of the extended query is provided as three tasks by this invention, i.e., extension of a query, processing of a query, and a rank of a result.

[0066]First, extension of a query is explained. Drawing 6 shows the example of extension of the query under the conventional query extension technique. The query of "searching a document including the word of car and dealer" is corrected, and the word relevant to car and dealer is added. A semantically similar related word and the related word which has syntactic cooccurrence relation are determined using the table shown in drawing 3. The example of the query extension by the origin of the query extension technique which has two or more degrees of fragmentation depended on this invention is shown in drawing 7. The extended technique of the query which has two or more degrees of fragmentation changes the word of car and dealer into the concepts Sem1 and Sem2 using the table shown in (a) of drawing 3. Using the table showing a word in (c) of drawing 4 after [corresponding to the word] changing into the semantic concept of an upper level more, the semantic concept is extended so that syntactic relations may be included, and the proper noun in the original query is extended so that the related word from a coincidence table may be included.

[0067]When the query Q including both the word in a dictionary and the word which is not in a dictionary is given, Q is expressed by the following formulas (15).

[0068]

$$Q=(s_1{}^{**} \dots {}^{**}s_m)^{**}(p_1{}^{**} \dots {}^{**}p_n) \dots (15)$$

s_i expresses the word in a dictionary with a formula (15), and p_j expresses with it the word which is not in a dictionary. There are n words which does not have a word in a dictionary in those with m piece and a dictionary in the query Q. If such a query is given, query extension technique which has two or more degrees of fragmentation will be performed as follows.

[0069]1. every in Q -- it corresponds to the s_i obtained from the table showing s_i (i= 1, ..., m) at (a) of drawing 3 -- replace by the semantic concept of an upper level more. C_i [each of the concept replaced in this way] is written.

[0070]2. every obtained at Step 1 -- extend Q by searching for and adding the word which has syntactic relation to C_i (i= 1, ..., m) using the table shown in (c) of drawing 4. A S-S type entry contributes to the addition of a concept, and a S-P type entry contributes to a proper noun.

[0071]3. Extend Q by searching for and adding the word to produce and which has syntactic relation using the table shown in (c) of drawing 4 with each p_j (j= 1, ..., n). A P-S type entry contributes to the addition of a concept, and a P-P type entry contributes to a proper noun.

[0072]4. Remove the word or the concept of a redundant query from Q.

[0073]Compared with the query extended by the conventional technique, the query extended by this invention is compacter and there are few items which should be checked. That is because the word of the query is changed into the entity in the coarser degree of fragmentation. As a result, the cost of the query processing of the query extended by this invention will become still smaller. Next, the number of the entities (a word or concept) introduced in the query extension which has two or more degrees of fragmentation of query

extension of a Prior art and this invention estimates. As mentioned above, the mean number of the word in a dictionary by which grouping was carried out more under the semantic concept of an upper level is expressed with f. Here, the mean number of the proper noun which has syntactic relation which was semantically related with a word and which set the mean number of the concept of an upper level to g more, and was related with a word is set to h. Then, the number of the words in Q under extension (BQ) of a fundamental query is the expansive sum total by which it is generated at Steps 1, 2, and 3, as shown in the following formulas (16).
[0074]
Numbers-of-words [BQ] =(mf)+m(g+h)+n (g+h) .. . (16)
Here, since each of m words in a dictionary is replaced by f semantically similar words, the 1st paragraph is produced. Since the individual (g+h) addition of the coincidence word in a dictionary and the coincidence word which is not in a dictionary is carried out to each of m words in a dictionary, the 2nd paragraph is produced. The 3rd paragraph corresponds to each of n proper nouns which added the coincidence word of the individual (g+h). Similarly, the number of the word in Q under the query extension (MGQ) which has two or more degrees of fragmentation, and concepts is expressed with the following formulas (17).
[0075]
Numbers of words [MGQ] =m+m(g/f+h) +n (g/f+h) ... (17)
Since a semantic expression of the upper level is here used more about the group of the similar word currently used, that the compression element f appears is a point which is greatly different substantially. Therefore, the number of the word/concepts which are included in a query has decreased in the meaning stricter than what is depended on a fundamental query extension technique by the query extension technique which has two or more degrees of fragmentation. In the table of (c) of drawing 4, if the number of the proper nouns for every word is small, the complexity of the query by the technique of this invention will be reduced based on the element f.
[0076]Shortly, query processing is explained. In the query processing based on the conventional strict matching, shortly after it turns out that the conditions about the predicate of the search relevant to a query are not fulfilled, retrieval processing will be ended. Since search is due to similarity, in actual IR, that is not right. Especially a user is going to look at the result by which that it is also partial matched the user's search criteria. Therefore, to the query which has N words, N times of lookups are required and this is not dependent on the Boolean conditions in the predicate of search. Since partial matching is supported, it is necessary to add rank processing after query processing. Rank technique needs the information about the frequency within which word of the documents matches a query, and the document of the word.
[0077]Here, in two techniques, it analyzes about the cost of the lookup at the time of processing a query. There is a fundamental difference in a cleanup cost for two factors. These two factors are shown below.

[0078]- There are more numbers of words in fundamental query extension than the numbers of words in the query extension which has two or more degrees of fragmentation.
[0079]- Differ by the technique to which a lookup is performed and whose number of entries of each table is two.
[0080]Here, the lookup cost of the query Q mentioned above is estimated. the table is systematized with a balanced search structure and lookup operation of a table responds to the number of lines of a table -- logarithm -- it is assumed that it changes-like. Therefore, the lookup cost at the time of performing Q in fundamental query extension using the above-mentioned estimated type becomes as shown in the following formulas (18) and (19).
[0081]
Lookup cost. (Q, BQ) = mf-log (number of lines [2 (b)]+(m+n), (g+h), and log (the number of lines [3 (b)]) ... [(18) = mf-log(W+V)+(m+n), (g+h), and log (V(V-1) /2+VW+W(W-1) /2)] ... (19))
[0082]The lookup cost at the time of similarly performing Q in the query extension which has two or more degrees of fragmentation becomes as shown in the following formulas (20) and (21).
[0083]
Lookup cost (Q, MGQ) =. m-log (the number of lines [4 (b)] + (m+n) -(g/f+h)- log (the number of lines [4 (c)])) ... (21).
[0084]Since the size of two tables where the number of times of the lookup of the word in a dictionary decreases by the element f in MGQ, and is the execution target of a lookup becomes small, it is clear that the cost of the query processing in MGQ becomes smaller than the cost in BQ.
[0085]Next, how to rank this invention is explained. In a query processing stage, expression of the word in the coarser degree of fragmentation is used for removing an unrelated document. However, since they fulfill two conditions, i.e., the conditions that "car" and "dealer" are included in a coarser fragmentation degree level, the document which serves as a candidate has the same rank. This is not preferred as a result of query processing. Therefore, in the stage of a rank, the word of the origin in the document which serves as a candidate is accessed, and it is used for a rank.
[0086]The document which has a keyword which fulfills the following conditions and which becomes four candidates is shown by drawing 8.
[0087]Conditions: (Sem1 ** Ford ** Buick) ** (Sem2 ** Ford** BUICK).
[0088]The first matching keyword is searched for a rank. Therefore, ("car", "dealer"), ("auto", "dealer"), ("auto", "sales office"), and ("Ford", "showroom") are used for ranking the grade of relevance.
[0089]The document which serves as a candidate is ranked based on the grade of the relaxation about the word which matched in the document which has a word in a query.
[0090]For example, the grade of relaxation is defined in order of E<Se<Sy<X (that is, with no strict matching < semantic relaxation < syntactic relaxation < matching). Here, the result of the

query using the word which relaxation was made on the higher level will contain that more nearly unrelated to a user about the word of a query. However, the order of the grade of relaxation and a definition are arbitrary by the requirements for application. The rank of the document which serves as a candidate becomes higher, so that smaller relaxation is used, although the document which serves as a candidate is looked for. The highest rank is given to the document which has a word of "car" and "dealer" in the lower part of drawing 8. This is because the candidate's word matched the word of the query strictly. The rank with a document expensive to the 2nd which has a word of "auto" and "dealer" is given. This is because semantic relaxation (that is, the term of a query is replaced with a semantically related term) is needed only for one word in order to make the word "car" of a query match. About other ranks, as shown in drawing 8, it is carried out.

[0091]Rank technique is performed based on the following two standards.

[0092]- The relation between keyword Word1 in Q and document Doc1, Word2 in Doc2, Word3 in Doc3, and Word4 in Doc4 about the keyword of the given query Q, respectively, When you have strict matching, matching by semantic query relaxation, matching by syntactic query relaxation, and no matching, a document is ranked in order of Doc1>Doc2>Doc3>Doc4.

[0093]-. Correspond to M documents, Doc_i (i= 1, ..., M), and document Doc_i, respectively. The rank (score) about the number of keywords and $Match_i$ (i= 1, ..., M) which match a query, $Match_1 > Match_2 > Match_3$ ... When it is $Match_{M-1} > Match_M$, it is $Doc_1 > Doc_2 > Doc_3$... It becomes $Doc_{M-1} > Doc_{M0}$.

[0094]If based on the rank technique which uses the query provided with two keywords and which was mentioned above, the two-dimensional rank graph in the case of searching a document with the query which has two words as shown in drawing 9 will be generated. If a query is not extended, only the document within a slot (E, E) will be searched. if both semantic extension of a query and syntactic extension are used, unless a document will be [slot (X, X)] alike, all the related documents are searched.

[0095]This rank graph is expressed as a procession. A rank graph is expressed by the procession of Nx4, and M (i, j) (i= 0 ... N, j= 0...3) about the query which has N terms. For example, the rank graph of drawing 9 is expressed as the procession M (i, j) (i= 0...2, j= 0...3). For example, a slot (E, E), (Se, E), (Se, Sy), and (X, X) are expressed within the procession as a slot (3, 3), (2, 3), (2, 1), and (0, 0), respectively. According to this expression, each document can be ranked easily as follows.

[0096]- To the document within a slot (n, m), when m is from 0 to 3, the rank of these documents becomes a score higher than the document within a slot (i, j) (i= 0 ... n, j= 0...3).

[0097]- The score of the rank of the document within a slot (n1, m1) becomes more than the score of the document within a slot (n2, m2), when it is n1>=n2 and m1>=m2.

[0098]Expression of this rank graph is realized by the commercial visualization tool. For example, the visualization method called Cone Trees is changed by adding the depth about a

three-dimensional rank expression, and it deals in it. For details, April, 1993, Communications of the ACM, Vol. 36, No. 4, and page 57-71, G. Refer to "Information Visualization Using 3D Interactive Animation" by G. Robertson etc.

[0099]If based on this rank technique, the result within the slot of the upper part of drawing 9 is ranked by a score higher than the result in the lower part. However, it is difficult to rank the result of the slot which belongs to the same class in drawing 9. Drawing 10 shows how such a rank is performed. The slot shown as a result is further classified into a class, and it is made to have the rank whose slot of the same class is the same there.

[0100]Query processing by this invention is continuously performed for every class using the class structure shown in drawing 10. A user publishes a query with two keywords and the case where it is required that top 50 results should be searched is considered. If drawing 10 is referred to, a query processor may generate search results in the class 0 first. When there are more search results than 50, the query processor can end processing, without performing a query extension task. When the number of the search results in the class 0 is less than 50, the query processor can generate the result in the class 1 (for example, a slot (2, 3), and (3, 2)). When there are more totals of search results (for example, it can set in the class 0 and the class 1) than 50, a query processor ends processing, without carrying out query processing further. The query processor should be cautious of a slot (2, 3) and (3, 2) an inner result being continuously generable. That is, the query processor can generate the result of a slot (2, 3) first. When the total of search results exceeds 50, the query processor can end processing, without generating a result in a slot (3, 2). A query processor can continue generation of the further result from the remaining slot and class as mentioned above until the total of search results exceeds 50, or until it reaches the last class.

[0101]When it is changed by the user noting that one keyword is more important for the above-mentioned example than other keywords, an order that a query processor searches the slot of search results is corrected according to the change. For example, when a user specifies it that the keyword 1 is more important than the keyword 2, an order of the horizontal query processing in a class is drawn as shown in drawing 11. That is, in this example, a query processor generates search results into a slot (3, 2) first. Next, when the total of search results is less than 50, a query processor generates a result into a slot (2, 3) after that.

[0102]Drawing 12 shows the physical configuration of the system by which this invention is performed. Such a system contains the database 1206 which memorizes the aggregate of a document. This database contains the index 1208 for memorizing those relations to a concept (for example, semantic or syntactic concept) and the aggregate of a document. Further, a system generates the index 1208 and contains the indexer 1210 for generating the concept which has the degree of fragmentation of a higher rank more, and the index 1208 including those relations to the aggregate of a document. The processor 1204 is used for receiving the query specified by the user via the user interface 1202. Next, the processor 1204 processes a query and performs a rank function. It ranks with the result of a query and a function is again

displayed on a user via the user interface 1202.

[0103]The person skilled in the art can understand that operation of this invention is not what is restricted to the example illustrated by drawing 12. The person skilled in the art can actually acquire the same effect using other alternative hardware environment, without deviating from the range of this invention. For example, it performs by an element with separate **** and various functions (for example, it ranks with query processing and a function is performed by another component), or is performed by the single element (for example, a single processor performs index attachment, query processing, and a rank function).

[0104]In short, this invention has held the original validity (precision and recall) of the group of the keyword about the inputted document, the dictionary including a definition, and the query. Index attachment (saving of an index area) covering two or more degrees of fragmentation and a new technique for supporting extension of a query using query processing (saving of processing time) are provided effectively.

[0105]Since a query is simplified by the index attachment technique and query processing technique covering two or more degrees of fragmentation depended on this invention, the size of the index which shows the relation of a word becomes smaller, and the processing time of a query becomes short by them. Since the rank technique of this invention is based on a certain word from the beginning in a document, consistency is maintained at the result of a rank.

[0106]It is clear from the indication so far and instruction that a person skilled in the art can make other various change and corrections to this invention. Therefore, although this specification has described only some examples of this invention, various change can be considered to this invention, without deviating from the intention and range of this invention.

[0107]

[Effect of the Invention]According to this invention, in order to solve the problem of a word mismatch, and inefficiency [of the query processing produced as a result], the index of small size is used and efficient query extension is performed. It can specifically be semantically similar with the word specified in the query, the query can be extended physically and notionally using the word which has relation syntactically, and it can lessen missing a related document as a result.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

[Field of the Invention]This invention relates to the field of the index generally applied to collecting the documents in a database, and a query. It is related with reduction of the size of the index used for carrying out effective extension of a query, and processing and extension of a query in more detail, and processing of a continuous query.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

[Description of the Prior Art]The conventional search system which searches a document is based on the common principle and methodology which classify a document by applying a query. A document is specified a priori by an expert or the librarian, and index attachment is usually manually done using the adjusted term. Index attachment of the document may be carried out again based on the word (word) included in the document. A user connects between them with the word chosen from the term which can be specified with a suitable Boolean operator, and searches a document. A strict matching strategy is used in a such type system. Although this approach has many advantages of being simple and highly precise, the problem of a word mismatch produces it.
[0003]The author is the document, and the problem of the word mismatch in information retrieval is produced by using another word, when a certain word is being used and a user specifies the same concept as it as it in a query, although a certain concept is expressed. Drawing 1 shows that the words used in the document of HyperText Markup Language (HTML) related with "car (passenger car)" and "dealer (store)" may differ among various documents. Languages other than HTML like an extensible markup language (XML) and Standard Generalized Markup Language (SGML) are also used. When a user uses the word of "automobile (car)" and "dealer (store)" for a query, a result with which one cannot search the target document on the problem of a word mismatch is brought.
[0004]In this specification, since the object of search assumes that English is mainly contained, each element of the query used for search is described in English. However, these can also be expressed in the language of which country according to a user's demand. Here, the meaning in Japanese of the element will be expressed in a parenthesis (accepting necessity) following the element described in said English. Therefore, Japanese in the parenthesis concerned is for only explaining the meaning of the element of a query, and does not affect the result of a query.
[0005]Extension of the query is suggested as a technique which solves such a problem. The word (for example, word for which it has a related meaning of a synonym or others) to which

the meaning was [this approach] similar, and a syntactically related word. A query is extended by using (for example, the word group which appears simultaneously in the same document above fixed frequency is a syntactic coincidence word) as a word in a query. In this way, extension of a query will increase a possibility of matching the word in a related document. Use of extension of a query will extend a query including the word of "car dealer (store of a passenger car)" so that the term of the meaning same as follows may be included.

[0006]Line 1. [("car(passenger car)" OR"automobile (car)" OR"auto (car)" OR "sedan (sedan)") OR line 2.] ("Ford(Ford car)" OR "Buick (BYUIKKU vehicle)") AND line 3. ("dealer(store)" OR"Showroom (showroom)" OR "SalesOffice (sales store)").

[0007]There are two types of extension of the query contained in the above-mentioned example. Extension of the query of the line 1 and the line 3 adds the additional word relevant to "car" and "dealer" in a definition. That is, a semantically similar word is added. "automobile", "auto", and "sedan" are words which has a meaning similar to the word of "car." Similarly, "Showroom" and "SalesOffice" are words which has a meaning similar to the word of "dealer." Extension of the query of other types is shown in the line 2.

This is based on syntactic cooccurrence relation.

Many words used with World Wide Web (it is also only called a web) are proper nouns actually.

It is not found in a term dictionary.

For example, a proper noun is called Ford, Buick, NBA, and NFL (National Football League). As mentioned above, syntactic cooccurrence relation is drawn by analyzing the frequency where two words appears simultaneously in the same document. This is based on assumption that a possibility that those words are related is high, when two words appears in the same document frequently. as the word generated with "Ford" -- "dealer (store)" "body shop (a body factory", "Mustang (Mustang: name of the car by Ford Co.)", "Escort (escort: name of the car by Ford Co.)", etc. can be considered.)

[0008]In order to support extension of a query, the index of the word associated by the definition and a syntactic relation like coincidence information must be maintained appropriately. The index related with a word by the definition is constituted as a hierarchy cluster of a layered structure, a semantic network, or a related word. . About said layered structure, were carried out in Athens, Greece, in August, 1997. the 23rd International. Conference on VeryLarge. The page 538-547 of the proceedings of Data Bases, and W. Please refer to "Facilitating Multimedia Database Exploration through Visual Interfaces and Perpetual Query Reformulations" besides Li. About said semantic network, 1990 and International Journal of Lexicography 3 (4), G. A. Miller in the page 245-264 "Nouns in WordNet : Refer to A Lexical Inheritance System." About the hierarchy cluster of a related word. Refer to "The SMART and SIRE Experimental Retrieval Systems" by G. Salton of New York, McGraw-Hill, and the page 118-155, etc. in 1983. Since a syntactic relation like syntactic cooccurrence relation is expressed with binary relations, the size of a syntactic-related index is

dramatically large. Some techniques are proposed in order to solve this problem. The proceedings of the Fifteenth annual International ACM SIGIR Conference [in / techniques / such / 1992 and Denmark], G. "Use of syntactic context to produce term association lists for text retrieval" by Grefenstette, The proceedings of the 19 th Annual International ACM SIGIRConference in Zurich in 1996 and Switzerland, J. "Query Expantion Using Local and Global Document Analysis" by Xu etc., . It can set to Philadelphia, American Pennsylvania, in 1997. Refer to "Guessing Morphology from Terms and Corpora" by C. Jacquemin of the proceedings of the20 th Annual International ACM SIGIR Conference. Such a technique includes analysis of occurrence frequency, and use of a morphological rule (for example, all the words are changed into the gestalt used as the origin), or a term dictionary.

[0009]About the problem of a word mismatch, remarkable research has been done in the field of information retrieval (IR). About this, 1983 and McGraw-Hill BookCompany issue, G. "Introduction to Modern Information Retrieval" by Salton etc., 1989, Addison-Wesley Publishing Company, and Inc issue, G. It is based on Salton. "Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer", and 1997, Refer to "Readings in Information Retrieval" by K. Sparck Jones of San Francisco, American California, and Morgan Kaufmann, etc.

[0010]However, these the researches of most point to points about the standard of search, such as precision and recall. How to support extension of a query effectively (in 1993, the proceedings of the 3 rd Text Retrieval Conference in State Gaithersburg of Maryland) C. Although there are some researches which suggested the mechanism of refer to "Automatic Query Expansion Using SMART" or index attachment by Buckley etc., two problems without the solution to satisfy still remain. The 1st problem is a proper noun with many separate words in a set (for example, web) of a certain document.

Since many each words have the same word and the syntactically related word semantically, the size of an index is becoming very large.

Since a query is extended by the additional word, the 2nd problem is that the cleanup cost of a query becomes high.

[0011]Since the number of documents increases dramatically, and the word currently used is very various, and it is inconsistent and is occasionally wrong when dealing with the document information collected from the web (for example, type error), these problems become increasingly remarkable. In a certain research, almost all the user query about a web usually has two words. About this, they are proceedings of 1995 and Digital Libraries (DL'95), B. Croft etc. "Providing Government Information on the Internet : Please refer to Experienceswith THOMAS." However, if query extension is used, the length of a query will become long substantially. As a result, most existing search engines on a web can provide query expanded function.

[0012]Here, the existing research in the field of query extension is outlined. Query extension attracted remarkable attention in the field of IR. However, the portion which has attracted

attention until now was evaluating the grade of the standard (namely, precision and recall) of the search improved by extension of a query. Another research has focused on building a dictionary, in order to identify 1 set of similar terms about the word of the given query. However, old research is not tackling the point of making small the problem of efficient processing of a query when a query is extended, and size of the index used for performing extension and processing of a query. The problem which ranks a document based on strict matching and resemblance matching is left behind as a difficult thing.

[0013]SMART is one of the advanced information retrieval systems known well. About this, 1971, American New Jersey Englewood. The SMART Retrieval System -Experiments in Automatic Document Processing of the Gerard Salton edit published from Prentice-Hall of Cliffs, "Experiments with a fast algorithm for automatic classification" by R. T. Dattola of Chapter 12, And refer to "The SMART and SIRE Experimental Retrieval Systems" by G. Salton of the above-mentioned literature, etc. Each document is expressed in SMART by the terminological vector. Each position of the vector expresses the dignity (importance) of the corresponding term in a document. a set of the document of M individual which has N different terms is expressed with the procession of MxN. A query is also expressed as a terminological vector. Search of a document is due to calculation of the similarity corresponding to the cosine of query vectors and the vector of each document. INQUERY is among the systems known well [other]. About this, it is 3:327-332 of 1995 and Information Processing and Management, J. Refer to "Trec and tipster experiments with inquery" by Callan etc.

[0014]A potential meaning index (LSI) is the technique depending on the conceptual index by matching like a dictionary statistically drawn instead of individual term search. About this, 1990, Journal of the America Society of Information Science, "Indexing by latent semantic analysis" by R. Harshman of 41:391-407, etc., By and the proceedings of 1995 and the 1995 ACM Conference on Supercomputing. M. Please refer to "Computational Method for Intelligent Information Access" by W. Berry etc. It assumes that LSI has some structures which are not in sight, i.e., a potential structure, in directions of a word, and the structure needs to be exteriorized by analyzing generating of the word in a document. Therefore, a document is considered as a vector in the term space of the very big range, and each element of the vector expresses the occurrence frequency of the specific term in the given document. The standard based on the whole and local weighting refined more is also used, and it gets. The shortened singular value decomposition (SVD) evaluates the structure of word use covering a document. Please refer to the 2nd edition of "Matrix Computations" by G. Golub of Johns-Hopkins of American Maryland state Baltimore, etc. for this in 1989. Here, a search is performed using the database which has a singular value, and the vector acquired from shortened SVD. Let approach of this information retrieval be a standard coarser than the thing based on each term in the preliminary evaluation of LSI.

[0015]The automated query extension has been suggested for a long time as a technique which deals with a word mismatch problem. They are proceedings of the 17 th Annual

International ACM SIGIR Conference performed about this in Republic of Ireland Dublin in 1994, E. Please refer to "Query Expansion Using Lexical-Semantic Relations" by Voorhees. In a certain approach, a possibility that a query is extended using a thesaurus and a word matches within a related document is improved. In research, it turns out that an improvement has a limit only by using an only common thesaurus. Much innovative technique is also proposed. The proceedings of 1994 and the 3 rd International ConferenceonInformation and Knowledge Management, O. "Query Expansion Using Domain Adapted, Weighted Thesaurus in an Extended Boolean Model" by Kwon etc., The proceedings of the 16 th Annual International ACM SIGIR Conference performed in Pittsburgh, American Pennsylvania, in 1993, E. "Concept Based Query Expansion" by Voorhees, "Query Expansion Using Lexical-Semantic Relations" by E. Voorhees of the proceedings, And please refer to "Computational Methods for Intelligent Information Access" by M. W. Berry of the proceedings, etc. By the automated query extension, the increase in efficiency of 25% of search is calculated from 7% on the average as a result of the experiment. Please refer to "Automatic Query Expansion Using SMART" by C. Buckley of the proceedings, etc. for this.

[0016]Improvement of a query is attained also by including a syntactically related word. This approach carries out clustering of the word based on the coincidence information within a document, and extends a query using these clusters. Since this coincidence information is binary relations, the size of such an index will become always very big. A certain group used the compilation of the coincidence statistics about modification of a word, and changed or generated SUTEMA (stemmer), and it was proved [group] which is advantageous compared with the approach only using a morphological rule. Please refer to "Corpus-Specific stemming Using Word Form Co-occurrence" besides W. B. Croft of the proceedings of the Fourth Annual Symposium for this in 1994. Each above-mentioned technique which extends the term of a query to 1 set of semantically related terms is called whole (global) analysis. In query extension, the term from relevance feedback is also added to a query, and the efficiency of search is improved. June, 1990, Journal of the American. Refer to "Improving retrieval performance by relevance feedback" besides 41(4):288-297 of Society for Information Science, and G. Salton. This is called partial (local) analysis. By old research, by applying the whole analysis technique which used the context of a word and the structure of words and phrases to some groups of a document shows that search results more effective than simple local feedback and more positive are obtained. For details, refer to "Query Expansion Using Local and Global Document Analysis" by J. Xu of the above-mentioned literature, etc.

[0017]However, as mentioned above, it does not aim at that old research makes small size of the index used for solving the problem of efficient processing of a query when a query is extended, or performing query extension and query processing.

[Translation done.]

[Effect of the Invention]According to this invention, in order to solve the problem of a word mismatch, and inefficiency [of the query processing produced as a result], the index of small size is used and efficient query extension is performed. It can specifically be semantically similar with the word specified in the query, the query can be extended physically and notionally using the word which has relation syntactically, and it can lessen missing a related document as a result.

[Translation done.]

[Problem(s) to be Solved by the Invention]The purpose of this invention is to provide the method and device which perform efficient query extension using the index of small size, and process a continuous query, in order to solve the problem of a word mismatch, and inefficiency [of the query processing produced as a result]. It is semantically similar in more detail with the word specified in the query, the query is extended physically and notionally using the word which has relation syntactically, and it lessens missing a document related as a result.
[0019]In order to support extension of a query, the index of the word related about a definition and the word in syntactic cooccurrence relation needs to be maintained, and the following two problems become important about support of such query extension. The 1st is a problem of the size of an index table and the 2nd is a problem of the overhead of query processing. This invention also makes it the purpose to solve these problems.

[Translation done.]

[Means for Solving the Problem]According to this invention, a concept and treatment structure of information which consist of two or more degrees of fragmentation are used in order to support extension of a query. This invention contains an index attachment phase, a query processing phase, and a rank phase. In an index attachment phase, grouping of the semantically similar word is carried out as one concept, and one actual index size becomes small as a result for a semantic concept subdivided in this way more coarsely. A word between query processings and in a query uses the contents of a dictionary and actual data, it is mapped by a corresponding semantic concept and syntactic extension, and logical extension is performed to the original query as a result. An overhead about processing is avoided. Next, a word of the first query is used for ranking a document obtained as search results based on strict matching, semantic matching, and syntactic matching, and is used also for performing processing of a continuous query.
[0021]
[Embodiment of the Invention]The method for extending a query efficiently and the suitable embodiment of a device which are depended on this invention are described in detail below with an accompanying drawing. Although the following explanation is made about the PERICO object oriented database managerial system (OODBMS) of NEC, it should be cautious of this invention not being what is restricted to this. This invention is applied to the aggregate of various database systems and a document, and it deals in it.
[0022]This invention provides effective index attachment and processing support about extension of a query by introducing the concept of two or more degrees of fragmentation. The approach of this invention sets up an index about the word which is semantically similar after stemming (stemming) of a word using an available technique, and a syntactically related word. About said technique, the proceedings of the 19 th AnnualInternational ACM SIGIR Conference in 1996, Switzerland, and Zurich, J. "Query ExpansionUsing Local and Global Document Analysis" besides Xu, And the proceedings of the 20 th Annual International ACM SIGIR Conference in 1997 and Philadelphia, American Pennsylvania, C. Refer to "Guessing

Morphology from Terms and Corpora" of Jacquemin. The approach of this invention makes size of an index small by being the degree of fragmentation of a high level more, and merging some entries (tuple) into one entry. The tuple with the information on the degree of fragmentation of a higher level is used for searching a related document between query processings. Then, the original word of a query is the finer degree of fragmentation, and since the document obtained as a result between query processings based on strict matching, semantically similar matching, and syntactically related matching is ranked, it is used. Maintaining the accuracy of the whole in a search mechanism by using the index and query processing technique which have two or more degrees of fragmentation, size of an index can be made small and quicker query processing can be realized.

[0023]It explains that first it is adapted with the notation of two or more degrees of fragmentation in relation to the conventional index attachment currently used by almost all IR system how. Next, the estimate about the overhead over the memory location in the case of performing index attachment which has two or more degrees of fragmentation about a set of a predetermined document is performed.

[0024]In order to search a given word easily from a document list, the conventional IR system holds an index and extracts the group of the word simultaneously related with the obtained document. In this case, the term of a "document" should be cautious of relating to the combination of a text, an image, or a text and an image.

[0025]Drawing 2 shows the example of the index. The table shown in (b) of drawing 2 is the index which transposed the table shown in (a) of drawing 2. At drawing 2, in order to explain easily, these indexes are shown in the form of the table. However, in a actual environment, the class of the upper layer of PERCIO OODBMS of NEC is used, for example. If the example of one query is taken, a user will use a word "car (passenger car)" and "dealer (store)" first, and if a query is created, IR system will take out a document list from the line to which the table of (b) of drawing 2 corresponds. In this case, the answer of a query serves as an intersection of the document list obtained from two lines. The approach against this IR is what supports only strict matching clearly, A related document including the term which has similar meanings, such as "automobile dealer (store of a car)", "car showroom (showroom of a passenger car)", or "automobile showroom (showroom of a car)", cannot be obtained. Query extension is used from the description [query] "car" and "dealer" in relation to the special utility extended to the description ("car" or "automobile") and ("dealer" or "showroom"). Although this approach is realizable, a remarkable overhead will be invited to query processing. The lookup of several times is needed about each of a word similar as especially semantically instead of 2 times of the lookups about the index table of (b) of drawing 2 as the word in the original query. A thesaurus tool like an on-line dictionary is required to extend the term of a query to them and a semantically similar term. From these observation, when this invention searches a set of a document, it provides the more effective method of supporting extension of a query.

[0026]As stated previously, in order to avoid the mismatch of user's vocabulary and the

author's vocabulary, the extension of a query based on the method of extending a query using the word to which a meaning is similar, and the word which has syntactic relations is needed.
[0027]Drawing 3 shows the data structure for which an addition is needed making extension of a query easy in the conventional IR system. Especially drawing 3 shows the table drawn from the on-line dictionary by which grouping is carried out to the concept to which each word is semantically similar, and including a definition. The table shown in drawing 3 is simplified for explanation. For example, the group "car (passenger car)" of a similar term, "auto (car)", "automobile (car)", and "sedan (sedan)" are expressed as one symbolic entity and sem1. Unlike the semantic resemblance based on a dictionary or a thesaurus, the syntactic relation to IR is determined by the collection of a document itself. Especially the coincidence information on a word is used for associating two words syntactically. Drawing 3 (b) has illustrated the index showing this information. With the auxiliary index of drawing 3, a fundamental query extension technique is supported in IR system by using the conventional IR index of drawing 2. Fundamentally, if a user's query is given, the word list of a query will be extended so that a semantically similar word and a syntactically related word may be included.
[0028]Although an above-mentioned method is used for processing of the query using extension of the query, in this approach, the overhead about processing will become high. According to this invention, the index structure of the addition which can process a query more efficiently is used. The fundamental way of thinking of the approach of this invention changes the index of drawing 2 and drawing 3 so that a query may be extended notionally. Namely, the list of the word of a query is not physically extended by including a semantically similar word and a syntactically related word in a list, A query is notionally extended for the word of a query by [the / related] changing for the semantic concept of an upper level, and a syntactic-related (for example, cooccurrence relation) word more. This brings about addition of the capacity overhead by an additional index structure. However, since a user's query is processed more efficiently, saving can be attained as the whole.
[0029]As mentioned above, in order to process the extended query, an index table is changed as shown in drawing 4. Especially the index table shown in (a) of drawing 4 is drawn from (a) of drawing 2 by transposing each word (it is not a peculiar name) to the word of the semantic concept of an upper level more. The index table shown in (b) of drawing 4 is obtained in the word shown in (b) of drawing 2 by [to which they correspond] combining with the word of the semantic concept of an upper level more, and merging the entry of each document list. Therefore, the line entry corresponding to "car", "auto", "automobile", and "sedan" is expressed with (b) of drawing 4 as single entry Sem1. Similarly, the line corresponding to "dealer", "showroom", and "SalesOffice" of drawing 2 (b) is summarized to one line of a label called Sem2.
[0030]The index to a syntactically related word is usually quite larger than the index to a word semantically related from several reasons. Many words on a web are peculiar names, and is not found in a dictionary. In the experiment, when the document of 2,904 was analyzed, 42%

of keywords were found by WordNet. WordNet is an on-line dictionary which has 60,000 or more words. About this, 1990 and International Journal of Lexicography 3 (4), It is based on G. A. Miller of the page 245-264. "Nouns in WordNet : Please refer to A Lexical Inheritance System." 58% of word of the remainder includes the proper noun and the type error, and this has become the origin which hypertrophies the size of an index. In the conventional IR system, syntactic correlation is usually grasped by cooccurrence relation. Since the cooccurrence relation of the word within the same document is 1 to 1 relation, when n words is identified, the size of an index is set to (nx (n-1)) / 2 in the worst case. Carrying out index attachment of the cooccurrence relation of three or more words for the overhead of a huge memory location and index attachment requires cost dramatically.

[0031]The word (that which is semantically meaningful) found in the dictionary is set to S, and other words (proper noun) of all the are set to P. The cooccurrence relation between words is classified into three different categories based on the above-mentioned classification of the word in a dictionary, and the word which is not in a dictionary.

[0032]- P-P type: (name of Toyota cars), for example, (Toyota (Toyota), Avalon) (Acura (Aqura), Legend (name of the Aqura vehicle)) (Nissan (Nissan), Maxima (name of Nissan cars)),

[0033]- S-P type or P-S type:, for example, (Buick (name of Ford cars), car (passenger car)), Buick, dealer (store), (car, Ford) (Ford, auto (car)) (Ford, dealer) (Ford Co.),

[0034]- S-S type:, for example, (car, garage (garage)) (auto, garage),

[0035]Usually, it is difficult to change the P-P type entry which is not convertible for the coarser degree of fragmentation shown in (b) of drawing 3. However, other entries of all the have S word which can be replaced by the corresponding semantic concept of a higher level. The size of a coincidence index decreases and speedup of query processing is realized by this. Reduction of the size of an index is produced as follows. To each S-P type (w_i, X) entry, all the entry of (w_i, X) shown in (b) of drawing 3 is replaced by (Sem_i, X) of (c) so that w_i may correspond to semantic concept Sem_i. [of drawing 4] Here, the list of corresponding documents is also merged. The same procedure is applied also to a P-S type entry. As shown in (c) of drawing 4, an entry (Ford, car), and (Ford, auto) are replaced by (Ford, Sem1). Similarly, an entry (Ford, dealer), and (Ford, showroom) are replaced by (Ford, Sem2). Such a merge mechanism is explained using (a) of drawing 5, and (b).

[0036]A S-S type entry is merged by the following two methods.

[0037]- Single merge : merge of the type of many [one pair] / many pairs 1 as shown in (a) of drawing 5, and (b). For example, an entry (car, dealer), (automobile, dealer), and (auto, dealer) are replaced by (Sem1, dealer). The algorithm used here is the same as what is used with a S-P type and a P-S type.

[0038]- Compound merge : merge of a many to many type as shown in (c) of drawing 5. For example, an entry (car, dealer), (automobile, showroom), and (auto, SalesOffice) are replaced

by (Sem1, Sem2). The algorithm of this type of merge is as follows.

[0039]1. -- to each S-S type entry (w_i, X), all the entries [of drawing 3] of (b) of (w_i, X) are shown in (c) of drawing 4 so that w_i may correspond to semantic concept Sem_i -- as (Sem_i, X) -- it replaces.

[0040]To each entry of the type of 2. (Sem_i, w_j), w_j replaces (Sem_i, w_j) of such all by (Sem_i, Sem_j) so that it may correspond to semantic concept Sem_j.

[0041]The above-mentioned step 2 should be cautious of the ability to also perform before the above-mentioned step 1. It is carried out repeatedly and deals in Step 1 and Step 2 of this algorithm until what is merged is lost.

[0042]If two or more entries are merged, the syntactic word list of each entry will also be merged by the merger (UNION) operation according to it.

[0043]Index attachment technique which has two or more degrees of fragmentation is mounted in the upper layer of OODBMS, and it deals in it. In such mounting, the table shown in (a) of drawing 2, (a) of drawing 3, and (c) of drawing 4 is a class which has the contents. Other tables are classes which have only a pointer. Updating to an index, deletion, and inserting operation are performed by OODBMS via the program which transmits between automatic-checking-and-continuous-monitoring maintenance or a class. Maintenance of an index which has two or more degrees of fragmentation is performed cumulatively, and reorganization is not needed.

[0044]Next, the estimate of the example by this invention is calculated besides the index based on the conventional word, taking into consideration the overhead of a memory location added since it is required to support the index table based on a semantic concept. As mentioned above, the table shown in drawing 4 is introduced for efficient query processing. First, calculation about the estimate of the memory location about the index used by the conventional IR system, i.e., the table shown in drawing 2, is performed. The number of the documents in a predetermined aggregate presupposes that it is D. The numbers of words (number after removing a stop word and a grouping word using word stemming) in a dictionary in the aggregate of the predetermined document are W, and set to V the number of the words which is not in a dictionary. The average of the numbers of words in the dictionary for every document is set to w, the average of the numbers of words which are not in a dictionary is set to v, and the average of the document number for every word is set to d. The size of an index is calculated based on the number of entries (namely, the number of lines), and the size (namely, the number of pointers) of the whole table. Each entry of the table should be cautious of being expressed as pointer data. When these parameters are given, the size of the table shown in (a) of drawing 2 is expressed with the following formulas (2).

[0045]

The number of lines [2 (a)] =D ... (1)

Whole size [2 (a)] =(1+v+w) D ... (2).

[0046]Although the word where one pointer is needed for discernment of a document and which is not in a dictionary in the list of a word in each line is expressed, Although v pointers are needed on an average and also the word which is in a dictionary in the list of a word is expressed, since w pointers are needed on an average, it should be cautious of the paragraph of (1+v+w) having arisen. Similarly, the size of the table shown in (b) of drawing 2 is expressed with the following formulas (4).

[0047]

The number of lines [2 (b)] =W+V ... (3)

Whole size [2 (b)] = (1+d) and (W+V) ... (4).

[0048]Each line of this table is an average and needs d pointers which serve as an identifier of a document within a document list, and one pointer which points out the word itself.

[0049]Next, the memory location overhead of an on-line dictionary and a syntactic coincidence table required to support fundamental query extension is estimated. It is considered as the compression element obtained by carrying out grouping of the word which is in a dictionary about f to a semantic concept. Therefore, f can be regarded as a number of a word of averages by which grouping was carried out to one concept. The size of the table shown in (a) of drawing 3 can be expressed like the following formulas (6).

[0050]

The number of lines [3 (a)] =W/f ... (5)

Whole size [3 (a)] =W+W/f ... (6).

[0051]Since the memory location of the word in a dictionary is compressed based on the compression element f, a formula (5) is expressed in this way. It is shown that a formula (6) needs W pointers to express the word in the list of a word, and it is required for the pointer of a W/f individual to express a semantic identifier. The size of the table shown by (b) of drawing 3 is expressed with the following formulas (8) when the worst.

[0052]

The number of lines [3 (b)] =V(V-1) /2+VW+W(W-1) /2 ... (7)

Whole size [3 (b)] =(1+2+q) - (V(V-1) /2+VW+W(W-1) /2) ... (8).

[0053]In a formula (7), the 1st paragraph corresponds to the cooccurrence relation of a P-P type word, the 2nd paragraph corresponds to a S-P type or a P-S type, and the last paragraph corresponds to S-S type cooccurrence relation. q expresses the mean number of the entry in a document list for every paragraph showing cooccurrence relation. Since a syntactic term identifier is expressed, three pointers are needed, and two words is included in the cooccurrence relation of each line.

[0054]Next, it estimates about the memory location overhead about index attachment which has two or more degrees of fragmentation based on this invention which carries out grouping of the group of a semantically similar term to one unique semantic concept. As mentioned above, in order to calculate the size of the index table shown in drawing 4, it is necessary to estimate the average document number for every semantic concept, and the mean number of

the semantic concept for every document. It can be shown that the mean number of the document for every semantic concept becomes larger than d, and this extension does not become f-d since two or more terms are cut down by one semantic concept. On the other hand, the mean number of the concept for every document does not exceed w. It can be shown that this number actually becomes a thing similar to w. Based on these parameters, the memory location overhead of an addition of index attachment which has two or more degrees of fragmentation is calculable. The calculation about the table shown in (a) of drawing 4 is as follows.

[0055]

The number of lines [4 (a)] =D ... (9)

Whole size [4 (a)] =(1+v+w) D ... (10).

[0056]That is, size is the same as the table shown in (a) of drawing 2. On the other hand, the size of the table shown in (b) of drawing 4 is as follows.

[0057]

The number of lines [4 (b)] =W/f ... (11)

Whole size [4 (b)] = (1+df) and W/f ... (12).

[0058]Since a word is unified by the semantic concept, the number of the entries of the word in a dictionary decreases based on the element f. However, only the almost same part as the element increases the number of the documents for every semantic concept. As a result, the size of this table becomes the same thing as the table shown in (b) of drawing 2. In the degree of fragmentation of a higher level, the table indicated to be (a) of drawing 4 to (b) should be cautious of it being a table indicated to be (a) of drawing 2 to (b), respectively. Finally, the estimate of the memory location of the table shown in (c) of drawing 4 is calculated as shown in the following formulas (13) and (14).

[0059]

The number of lines [3 (b)] =V(V-1) /2+V, and(W/f)+ (W(W-1) /2f^2) ... (13)

Whole size [3 (b)] =(1+2+q) -V(V-1) /2+ (1+2+qf)

- Vand(W/f)+ (1+2+qf) (W(W-1) /2f^2) ... (14)

[0060]Fundamentally, S-S types, S-P types, or all the P-S type coincidence terms are compressed based on the element f, and it becomes small capacity substantially compared with the table shown in (b) of drawing 3.

[0061]Eventually, according to this invention, it is needed except the table shown in (b) of drawing 3. On the other hand, a fundamental query extension technique needs all the tables shown in drawing 2 and drawing 3. Therefore, although only the increment of the memory location of the table showing the cost about the memory location at the time of adopting the method of this invention in (b) with (a) of drawing 4 becomes large, since the size of the table shown in (c) of drawing 4 becomes small, it compensates for a part for said increase in cost selectively. It depends for the exact numerical value of saving on the value of various

parameters mentioned above. Even when the worst, an additional memory location is quite smaller than the twice of the memory location at the time of using a fundamental query extension technique.

[0062]The above-mentioned index attachment technique assumes having only a meaning with a single word, and it argues about it.However, a word usually has two or more meanings. For example, the word of "bank" is interpreted as a financial institution (bank) or a riverside. In order to take into consideration about the word which has two or more meanings, the word (shown by drawing 3) of a semantic word list shall belong to two or more conceptual numbers shown in (a) of drawing 4. For example, "bank" shall be related with Sem10 and Sem20. In order to perform extension of a query in consideration of two or more such meanings, when a query includes one word belonging to several different conceptual numbers, each of the different conceptual number should be taken into consideration in the case of processing of a query.

[0063]In the above-mentioned explanation, index technique is mounted in the upper layer of OODBMS of NEC, and the word in a semantic word list is related with the conceptual number by the pointer. Redundant data is not memorized but its cost of the memory location about a pointer is very low. WordNet provides the synonym to a certain word by the interpretation of various meanings, and ranks it according to the frequency where the meaning is used. For example, more "bank(s)" is interpreted as a financial institution rather than a riverside. The most general semantic interpretation is used for the present execution. However, a data structure is also extensible.

[0064]The grouping of a meaning except having been stated above can be taken into consideration. Only extension of the query by a synonym is taken into consideration in drawing 4. Relaxation of meanings of other molds, such as ISA and IS_PART_OF, can also be taken into consideration. Two or more tables of the gestalt shown in (a) of drawing 4 are generable about the grouping (one related with IS_PART_OF one about ISA for example) of various meanings. One table can also be used about the grouping of various meanings. When extending a query by both the synonym and a hypernym, a lookup is performed to two or more tables.

[0065]In order to cope with the problem of a word mismatch, the query processing technique needs to extend the word of a query using a related word. As a result, by the relevance over the word of the original query, the additional task which ranks a document is performed and it gets. Next, processing of the extended query is provided as three tasks by this invention, i.e., extension of a query, processing of a query, and a rank of a result.

[0066]First, extension of a query is explained. Drawing 6 shows the example of extension of the query under the conventional query extension technique. The query of "searching a document including the word of car and dealer" is corrected, and the word relevant to car and dealer is added. A semantically similar related word and the related word which has syntactic cooccurrence relation are determined using the table shown in drawing 3. The example of the

query extension by the origin of the query extension technique which has two or more degrees of fragmentation depended on this invention is shown in drawing 7. The extended technique of the query which has two or more degrees of fragmentation changes the word of car and dealer into the concepts Sem1 and Sem2 using the table shown in (a) of drawing 3. Using the table showing a word in (c) of drawing 4 after [corresponding to the word] changing into the semantic concept of an upper level more, the semantic concept is extended so that syntactic relations may be included, and the proper noun in the original query is extended so that the related word from a coincidence table may be included.

[0067]When the query Q including both the word in a dictionary and the word which is not in a dictionary is given, Q is expressed by the following formulas (15).

[0068]

$Q=(s_1{}^{**} \ldots {}^{**}s_m)^{**} (p_1{}^{**} \ldots {}^{**}p_n) \ldots (15)$

s_i expresses the word in a dictionary with a formula (15), and p_j expresses with it the word which is not in a dictionary. There are n words which does not have a word in a dictionary in those with m piece and a dictionary in the query Q. If such a query is given, query extension technique which has two or more degrees of fragmentation will be performed as follows.

[0069]1. every in Q -- it corresponds to the s_i obtained from the table showing s_i (i= 1, ..., m) at (a) of drawing 3 -- replace by the semantic concept of an upper level more. C_i [each of the concept replaced in this way] is written.

[0070]2. every obtained at Step 1 -- extend Q by searching for and adding the word which has syntactic relation to C_i (i= 1, ..., m) using the table shown in (c) of drawing 4. A S-S type entry contributes to the addition of a concept, and a S-P type entry contributes to a proper noun.

[0071]3. Extend Q by searching for and adding the word to produce and which has syntactic relation using the table shown in (c) of drawing 4 with each p_j (j= 1, ..., n). A P-S type entry contributes to the addition of a concept, and a P-P type entry contributes to a proper noun.

[0072]4. Remove the word or the concept of a redundant query from Q.

[0073]Compared with the query extended by the conventional technique, the query extended by this invention is compacter and there are few items which should be checked. That is because the word of the query is changed into the entity in the coarser degree of fragmentation. As a result, the cost of the query processing of the query extended by this invention will become still smaller. Next, the number of the entities (a word or concept) introduced in the query extension which has two or more degrees of fragmentation of query extension of a Prior art and this invention estimates. As mentioned above, the mean number of the word in a dictionary by which grouping was carried out more under the semantic concept of an upper level is expressed with f. Here, the mean number of the proper noun which has syntactic relation which was semantically related with a word and which set the mean number of the concept of an upper level to g more, and was related with a word is set to h. Then, the

number of the words in Q under extension (BQ) of a fundamental query is the expansive sum total by which it is generated at Steps 1, 2, and 3, as shown in the following formulas (16).
[0074]
Numbers-of-words [BQ] = (mf) +m(g+h) +n (g+h) ... (16)
Here, since each of m words in a dictionary is replaced by f semantically similar words, the 1st paragraph is produced. Since the individual (g+h) addition of the coincidence word in a dictionary and the coincidence word which is not in a dictionary is carried out to each of m words in a dictionary, the 2nd paragraph is produced. The 3rd paragraph corresponds to each of n proper nouns which added the coincidence word of the individual (g+h). Similarly, the number of the word in Q under the query extension (MGQ) which has two or more degrees of fragmentation, and concepts is expressed with the following formulas (17).
[0075]
Numbers of words [MGQ] =m+m(g/f+h) +n (g/f+h) ... (17)
Since a semantic expression of the upper level is here used more about the group of the similar word currently used, that the compression element f appears is a point which is greatly different substantially. Therefore, the number of the word/concepts which are included in a query has decreased in the meaning stricter than what is depended on a fundamental query extension technique by the query extension technique which has two or more degrees of fragmentation. In the table of (c) of drawing 4, if the number of the proper nouns for every word is small, the complexity of the query by the technique of this invention will be reduced based on the element f.
[0076]Shortly, query processing is explained. In the query processing based on the conventional strict matching, shortly after it turns out that the conditions about the predicate of the search relevant to a query are not fulfilled, retrieval processing will be ended. Since search is due to similarity, in actual IR, that is not right. Especially a user is going to look at the result by which that it is also partial matched the user's search criteria. Therefore, to the query which has N words, N times of lookups are required and this is not dependent on the Boolean conditions in the predicate of search. Since partial matching is supported, it is necessary to add rank processing after query processing. Rank technique needs the information about the frequency within which word of the documents matches a query, and the document of the word.
[0077]Here, in two techniques, it analyzes about the cost of the lookup at the time of processing a query. There is a fundamental difference in a cleanup cost for two factors. These two factors are shown below.
[0078]- There are more numbers of words in fundamental query extension than the numbers of words in the query extension which has two or more degrees of fragmentation.
[0079]- Differ by the technique to which a lookup is performed and whose number of entries of each table is two.
[0080]Here, the lookup cost of the query Q mentioned above is estimated. the table is

systematized with a balanced search structure and lookup operation of a table responds to the number of lines of a table -- logarithm -- it is assumed that it changes-like. Therefore, the lookup cost at the time of performing Q in fundamental query extension using the above-mentioned estimated type becomes as shown in the following formulas (18) and (19).
[0081]
Lookup cost. (Q, BQ) = mf-log (number of lines [2 (b)]+(m+n), (g+h), and log (the number of lines [3 (b)]) ... [(18) = mf-log(W+V)+(m+n), (g+h), and log (V(V-1) /2+VW+W(W-1) /2)] ... (19))
[0082]The lookup cost at the time of similarly performing Q in the query extension which has two or more degrees of fragmentation becomes as shown in the following formulas (20) and (21).
[0083]
Lookup cost (Q, MGQ) =. m-log (the number of lines [4 (b)] + (m+n) -(g/f+h)- log (the number of lines [4 (c)])) ... (21).
[0084]Since the size of two tables where the number of times of the lookup of the word in a dictionary decreases by the element f in MGQ, and is the execution target of a lookup becomes small, it is clear that the cost of the query processing in MGQ becomes smaller than the cost in BQ.
[0085]Next, how to rank this invention is explained. In a query processing stage, expression of the word in the coarser degree of fragmentation is used for removing an unrelated document. However, since they fulfill two conditions, i.e., the conditions that "car" and "dealer" are included in a coarser fragmentation degree level, the document which serves as a candidate has the same rank. This is not preferred as a result of query processing. Therefore, in the stage of a rank, the word of the origin in the document which serves as a candidate is accessed, and it is used for a rank.
[0086]The document which has a keyword which fulfills the following conditions and which becomes four candidates is shown by drawing 8.
[0087]Conditions: (Sem1 ** Ford ** Buick) ** (Sem2 ** Ford** BUICK).
[0088]The first matching keyword is searched for a rank. Therefore, ("car", "dealer"), ("auto", "dealer"), ("auto", "sales office"), and ("Ford", "showroom") are used for ranking the grade of relevance.
[0089]The document which serves as a candidate is ranked based on the grade of the relaxation about the word which matched in the document which has a word in a query.
[0090]For example, the grade of relaxation is defined in order of E<Se<Sy<X (that is, with no strict matching < semantic relaxation < syntactic relaxation < matching). Here, the result of the query using the word which relaxation was made on the higher level will contain that more nearly unrelated to a user about the word of a query. However, the order of the grade of relaxation and a definition are arbitrary by the requirements for application. The rank of the document which serves as a candidate becomes higher, so that smaller relaxation is used, although the document which serves as a candidate is looked for. The highest rank is given to

the document which has a word of "car" and "dealer" in the lower part of drawing 8. This is because the candidate's word matched the word of the query strictly. The rank with a document expensive to the 2nd which has a word of "auto" and "dealer" is given. This is because semantic relaxation (that is, the term of a query is replaced with a semantically related term) is needed only for one word in order to make the word "car" of a query match. About other ranks, as shown in drawing 8, it is carried out.

[0091]Rank technique is performed based on the following two standards.

[0092]- The relation between keyword Word1 in Q and document Doc1, Word2 in Doc2, Word3 in Doc3, and Word4 in Doc4 about the keyword of the given query Q, respectively, When you have strict matching, matching by semantic query relaxation, matching by syntactic query relaxation, and no matching, a document is ranked in order of Doc1>Doc2>Doc3>Doc4.

[0093]-. Correspond to M documents, Doc_i (i= 1, ..., M), and document Doc_i, respectively. The rank (score) about the number of keywords and $Match_i$ (i= 1, ..., M) which match a query, $Match_1 > Match_2 > Match_3$ ... When it is $Match_{M-1} > Match_M$, it is $Doc_1 > Doc_2 > Doc_3$... It becomes $Doc_{M-1} > Doc_{M0}$.

[0094]If based on the rank technique which uses the query provided with two keywords and which was mentioned above, the two-dimensional rank graph in the case of searching a document with the query which has two words as shown in drawing 9 will be generated. If a query is not extended, only the document within a slot (E, E) will be searched. if both semantic extension of a query and syntactic extension are used, unless a document will be [slot (X, X)] alike, all the related documents are searched.

[0095]This rank graph is expressed as a procession. A rank graph is expressed by the procession of Nx4, and M (i, j) (i= 0 ... N, j= 0...3) about the query which has N terms. For example, the rank graph of drawing 9 is expressed as the procession M (i, j) (i= 0...2, j= 0...3). For example, a slot (E, E), (Se, E), (Se, Sy), and (X, X) are expressed within the procession as a slot (3, 3), (2, 3), (2, 1), and (0, 0), respectively. According to this expression, each document can be ranked easily as follows.

[0096]- To the document within a slot (n, m), when m is from 0 to 3, the rank of these documents becomes a score higher than the document within a slot (i, j) (i= 0 ... n, j= 0...3).

[0097]- The score of the rank of the document within a slot (n1, m1) becomes more than the score of the document within a slot (n2, m2), when it is n1>=n2 and m1>=m2.

[0098]Expression of this rank graph is realized by the commercial visualization tool. For example, the visualization method called Cone Trees is changed by adding the depth about a three-dimensional rank expression, and it deals in it. For details, April, 1993, Communications of the ACM, Vol. 36, No. 4, and page 57-71, G. Refer to "Information Visualization Using 3D Interactive Animation" by G. Robertson etc.

[0099]If based on this rank technique, the result within the slot of the upper part of drawing 9 is ranked by a score higher than the result in the lower part. However, it is difficult to rank the

result of the slot which belongs to the same class in drawing 9. Drawing 10 shows how such a rank is performed. The slot shown as a result is further classified into a class, and it is made to have the rank whose slot of the same class is the same there.

[0100]Query processing by this invention is continuously performed for every class using the class structure shown in drawing 10. A user publishes a query with two keywords and the case where it is required that top 50 results should be searched is considered. If drawing 10 is referred to, a query processor may generate search results in the class 0 first. When there are more search results than 50, the query processor can end processing, without performing a query extension task. When the number of the search results in the class 0 is less than 50, the query processor can generate the result in the class 1 (for example, a slot (2, 3), and (3, 2)). When there are more totals of search results (for example, it can set in the class 0 and the class 1) than 50, a query processor ends processing, without carrying out query processing further. The query processor should be cautious of a slot (2, 3) and (3, 2) an inner result being continuously generable. That is, the query processor can generate the result of a slot (2, 3) first. When the total of search results exceeds 50, the query processor can end processing, without generating a result in a slot (3, 2). A query processor can continue generation of the further result from the remaining slot and class as mentioned above until the total of search results exceeds 50, or until it reaches the last class.

[0101]When it is changed by the user noting that one keyword is more important for the above-mentioned example than other keywords, an order that a query processor searches the slot of search results is corrected according to the change. For example, when a user specifies it that the keyword 1 is more important than the keyword 2, an order of the horizontal query processing in a class is drawn as shown in drawing 11. That is, in this example, a query processor generates search results into a slot (3, 2) first. Next, when the total of search results is less than 50, a query processor generates a result into a slot (2, 3) after that.

[0102]Drawing 12 shows the physical configuration of the system by which this invention is performed. Such a system contains the database 1206 which memorizes the aggregate of a document. This database contains the index 1208 for memorizing those relations to a concept (for example, semantic or syntactic concept) and the aggregate of a document. Further, a system generates the index 1208 and contains the indexer 1210 for generating the concept which has the degree of fragmentation of a higher rank more, and the index 1208 including those relations to the aggregate of a document. The processor 1204 is used for receiving the query specified by the user via the user interface 1202. Next, the processor 1204 processes a query and performs a rank function. It ranks with the result of a query and a function is again displayed on a user via the user interface 1202.

[0103]The person skilled in the art can understand that operation of this invention is not what is restricted to the example illustrated by drawing 12. The person skilled in the art can actually acquire the same effect using other alternative hardware environment, without deviating from the range of this invention. For example, it performs by an element with separate **** and

various functions (for example, it ranks with query processing and a function is performed by another component), or is performed by the single element (for example, a single processor performs index attachment, query processing, and a rank function).

[0104]In short, this invention has held the original validity (precision and recall) of the group of the keyword about the inputted document, the dictionary including a definition, and the query. Index attachment (saving of an index area) covering two or more degrees of fragmentation and a new technique for supporting extension of a query using query processing (saving of processing time) are provided effectively.

[0105]Since a query is simplified by the index attachment technique and query processing technique covering two or more degrees of fragmentation depended on this invention, the size of the index which shows the relation of a word becomes smaller, and the processing time of a query becomes short by them. Since the rank technique of this invention is based on a certain word from the beginning in a document, consistency is maintained at the result of a rank.

[0106]It is clear from the indication so far and instruction that a person skilled in the art can make other various change and corrections to this invention. Therefore, although this specification has described only some examples of this invention, various change can be considered to this invention, without deviating from the intention and range of this invention.

[Translation done.]

[Brief Description of the Drawings]
[Drawing 1]It is a figure showing the problem of the word mismatch about information retrieval.
[Drawing 2]It is a figure showing the example of the index currently used conventionally with the information retrieval system of strict matching.
[Drawing 3]In order to use it with the conventional information retrieval system, it is a figure showing the example of the index obtained by carrying out grouping of the word to a semantically similar concept and syntactically related extension.
[Drawing 4]In this invention, it is a figure showing an index structure required in order to perform query processing more efficiently.
[Drawing 5]It is a figure showing the processing which merges the entry of the index of a coincidence word.
[Drawing 6]It is a figure showing the query extension processing in the conventional information retrieval system.
[Drawing 7]It is a figure showing the query extension processing using the query extension technique which has two or more degrees of fragmentation depended on this invention.
[Drawing 8]It is a figure by this invention showing rank processing.
[Drawing 9]It is a two-dimensional graph showing the rank of the query which has two words.
[Drawing 10]It is a figure showing an order of continuous query processing.
[Drawing 11]It is a figure showing an order of continuous query processing in the case of being assigned to the importance of the level with a keyword.
[Drawing 12]This invention is a figure showing the physical configuration of one feasible embodiment.
[Description of Notations]
1202 User interface
1204 Processor
1206 Database
1208 Index

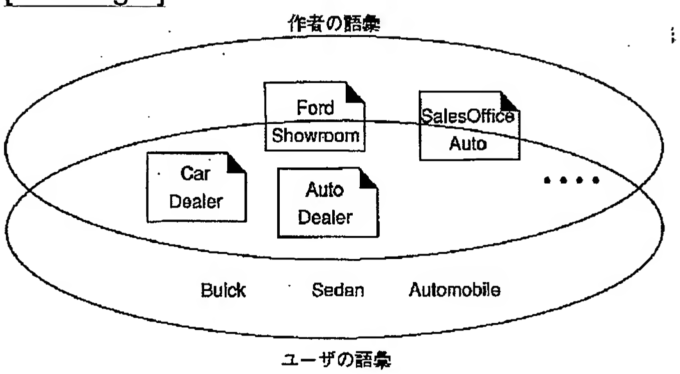
1210 Indexer

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

[Drawing 1]



[Drawing 2]

(a)

文書番号	ワードリスト
Doc1	Ford,Showroom
Doc2	Auto,SalesOffice
Doc3	Car,Dealer
Doc4	Auto,Dealer
:	:

(b)

ワード番号	文書リスト
Ford	Doc1
Showroom	Doc1
Auto	Doc2,Doc4
Dealer	Doc3,Doc4
:	:

[Drawing 3]

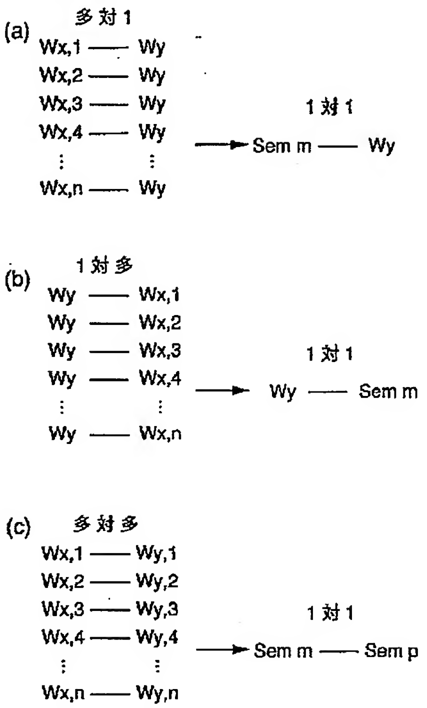
(a)

概念番号	意味的ワードリスト
Sem1	Car,Auto,Automobile,Sedan
Sem2	Dealer,Showroom,SalesOffice
Sem3	Garage,Parking
:	:

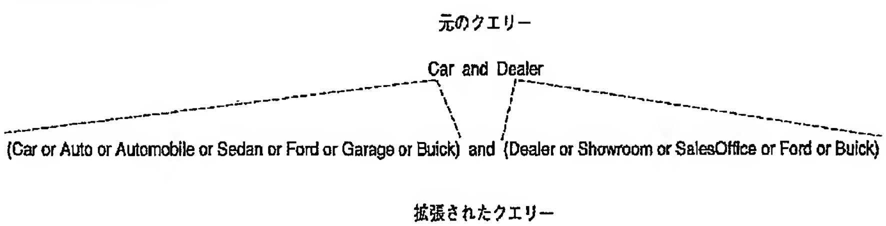
(b)

構文番号	構文的ワードリスト
Syn1	Buick,Car
Syn2	Car,Garage
Syn3	Auto,Garage
Syn4	Ford,Car
Syn5	Ford,Auto
:	:

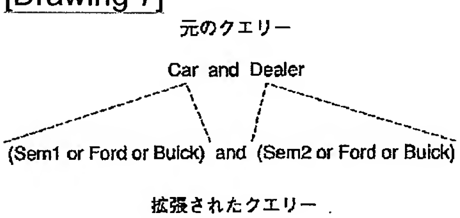
[Drawing 5]



[Drawing 6]



[Drawing 7]



[Drawing 4]

(a)

文書番号	概念・固有名称リスト
Doc1	Ford,Sem2
Doc2	Sem1,Sem2
Doc3	Sem1,Sem2
Doc4	Sem1,Sem2
:	:

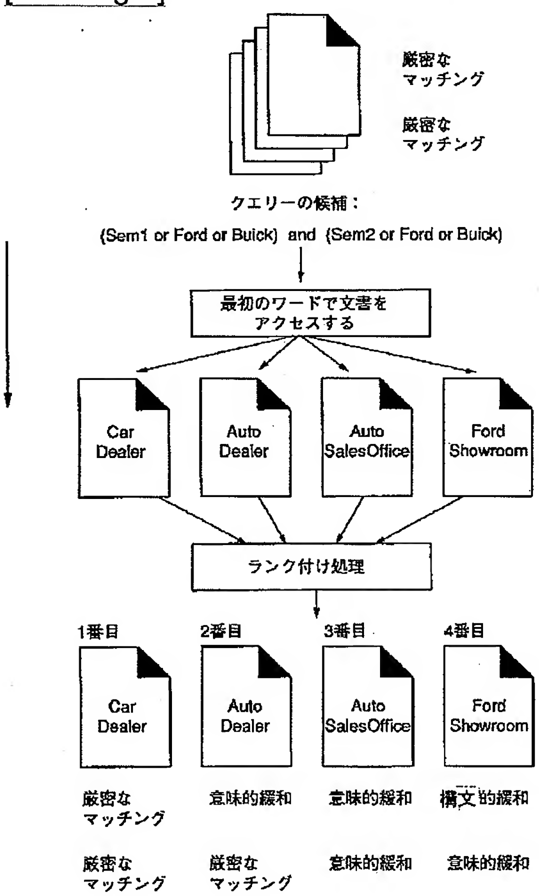
(b)

概念番号	意味別文書リスト
Sem1	Doc2,Doc3,Doc4
Sem2	Doc1,Doc2,Doc3,Doc4
:	:

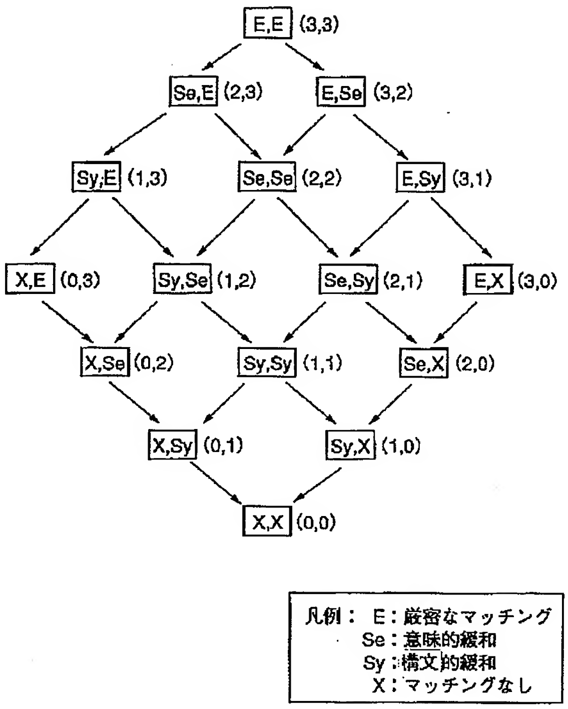
(c)

共起番号	構文別ワード・リスト
Syn1'	Buick,Sem1
Syn2'	Sem1,Sem3
Syn3'	Ford,Sem1
Syn4'	Sem7,Sem21
Syn5'	Ford,Buick
:	:

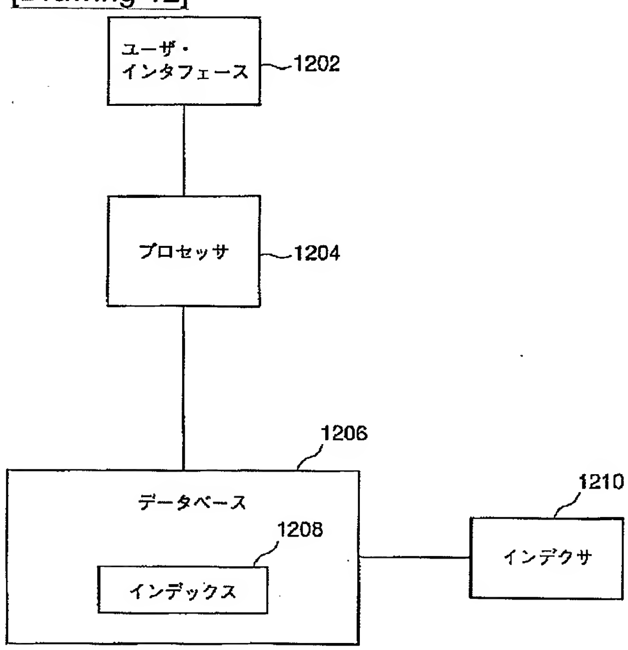
[Drawing 8]



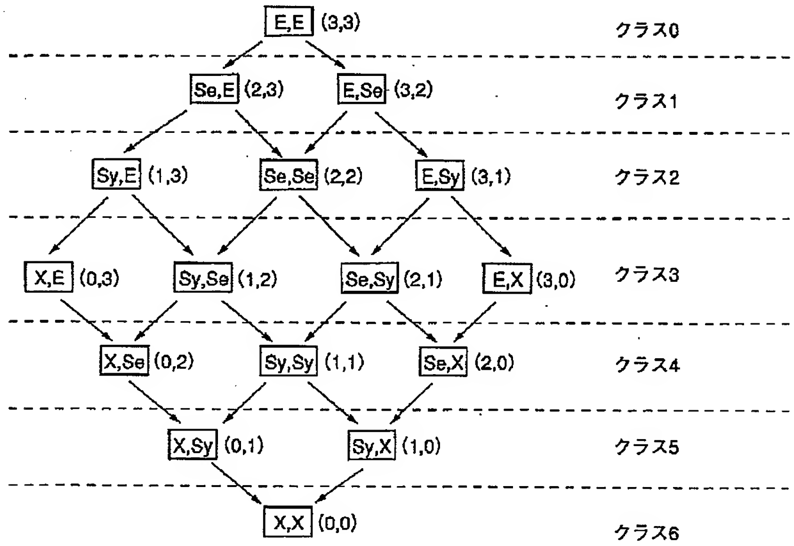
[Drawing 9]



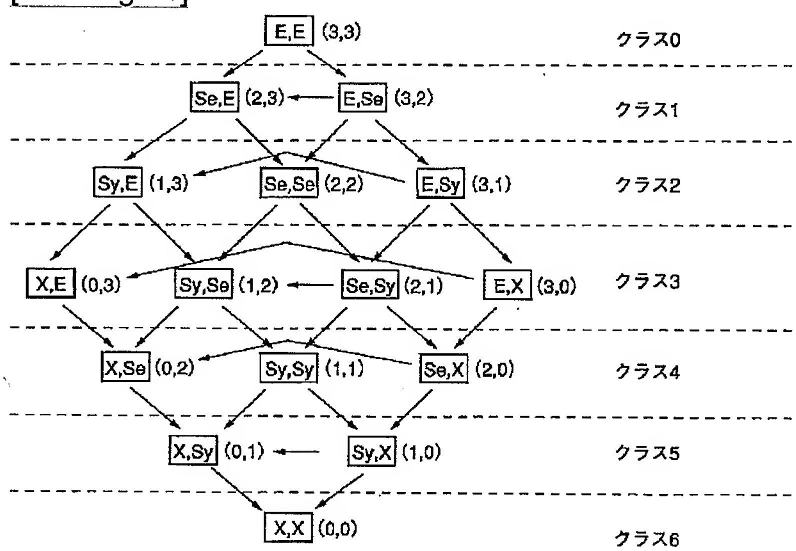
[Drawing 12]



[Drawing 10]



[Drawing 11]



[Translation done.]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開2000－137738

（P2000－137738A）

(43)公開日　平成12年５月16日(2000.5.16)

(51)Int.Cl.⁷	識別記号	ＦＩ		テーマコード*(参考)
Ｇ０６Ｆ　17/30		Ｇ０６Ｆ　15/403	３３０Ｂ	５Ｂ０７５
		15/40	３７０Ａ	
		15/403	３７０Ｚ	

審査請求　未請求　請求項の数76　ＯＬ　（全 19 頁）

(21)出願番号　特願平11－140695

(22)出願日　平成11年５月20日(1999.5.20)

(31)優先権主張番号　０９／１８５３２３

(32)優先日　平成10年11月３日(1998.11.3)

(33)優先権主張国　米国（ＵＳ）

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号

(72)発明者　ウェン　シャン　リー
アメリカ合衆国，カリフォルニア　95134，サンノゼ，110　ロブルス　アベニュー　エヌ・イー・シー・ユー・エス・エー・インク内

(74)代理人　100071272
弁理士　後藤　洋介　（外１名）

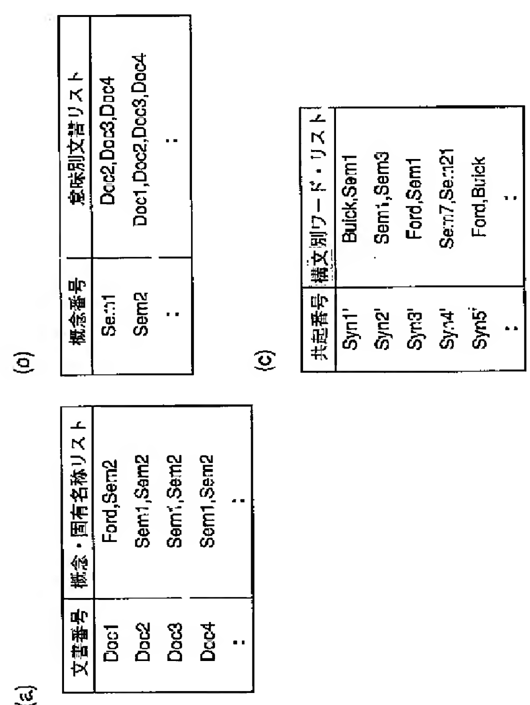
Ｆターム(参考)　5B075 ND03 NK02 NK35 NK49 NK54 PQ02 QM07 QP03 UU06

(54)【発明の名称】　複数の細分度のインデックス付けとクエリー処理を効果的に用いてクエリーの拡張を支援する方法、及び装置

(57)【要約】

【課題】　サイズの小さいインデックスを用いてクエリーを効果的に拡張し、連続的にクエリーを処理するための方法、及び装置を提供する。

【解決手段】　クエリーは、元のクエリーでユーザにより指定されたワードに意味的に類似するワード、及び構文的に関連するワードを用いて、概念的に拡張される。本発明は、効果的なクエリーの拡張を支援するために、複数の細分度を有する情報と、処理構造の概念を用い、インデックス付け、クエリー処理、及びランク付けの各処理を含む。



【特許請求の範囲】

【請求項１】　文書の予備的インデックス、文書内に含まれるワード、及び前記インデックスと前記ワードとの間の関係を含み、前記インデックス内のワードが元の細分度である、文書のデータベースを検索する方法であって、前記方法が、

ａ）小さなサイズの、より粗い細分度のインデックスを生成するために、前記予備的インデックスの中のワードを、対応する、より上位の概念に置き換えるステップと、

ｂ）元の細分度を有するクエリーのワードを、対応する、より上位の概念に置き換えることによって、文書のデータベースに適用される前記クエリーを論理的に拡張するステップと、

ｃ）より粗い細分度の前記インデックスを用いて、前記論理的に拡張されたクエリーを実行し、対応する、より上位の概念に関連する文書を検索するステップとを有することを特徴とする検索方法。

【請求項２】　請求項１において、

ｄ）関連性の順序に基づいて、検索された文書をランク付けするステップを、更に含むことを特徴とする検索方法。

【請求項３】　請求項２において、前記ランク付けステップで、検索された文書が、元の細分度を有するクエリーのワードを用いてランク付けされることを特徴とする検索方法。

【請求項４】　請求項３において、関連性の順序は、クエリーのワードと検索された文書に含まれるワードが、厳密にマッチする場合を始めとし、以降、意味的にマッチする場合、構文的にマッチする場合、マッチしない場合の順であることを特徴とする検索方法。

【請求項５】　請求項１において、前記置き換えステップで、より上位の概念が、より上位の意味的概念であることを特徴とする検索方法。

【請求項６】　請求項５において、より上位の意味的概念のそれぞれが、類義語を含むことを特徴とする検索方法。

【請求項７】　請求項１において、前記置き換えステップで、所定の基準を満たす予備的インデックス内のワードの一部だけが、より上位概念の対応するワードに置き換えられることを特徴とする検索方法。

【請求項８】　請求項７において、前記所定の基準は、前記ワードが用語辞書にあるかどうかに基づくことを特徴とする検索方法。

【請求項９】　請求項１において、前記置き換えステップで、より上位の前記概念が、より上位の構文的概念であることを特徴とする検索方法。

【請求項１０】　請求項９において、より上位の前記構文的概念のそれぞれが、あるレベルの頻度を越えて、文書内で共に発生するワードを含むことを特徴とする検索方法。

【請求項１１】　請求項１において、論理的にクエリーを拡張する前記ステップが更に、

ｂ）（ｉ）所定の基準を満たす、クエリーのワードのみを、より上位の意味的概念を有する、より上位の対応する概念に置き換えるステップを有することを特徴とする検索方法。

【請求項１２】　請求項１１において、論理的にクエリーを拡張する前記ステップが更に、

ｂ）（ｉｉ）対応する、より上位の前記概念のそれぞれに対して、構文的に関連するワードを付加することによって、前記クエリーを更に論理的に拡張するステップと、

ｂ）（ｉｉｉ）前記所定の基準を満たしていない、クエリー内のワードのそれぞれに対して、構文的に関連するワードを付加することによって、前記クエリーを更に論理的に拡張するステップとを有することを特徴とする検索方法。

【請求項１３】　請求項１２において、論理的にクエリーを拡張する前記ステップが更に、

ａ）（ｉｖ）所定の基準を満たす、構文的に関連する前記ワードを、関連する、より上位の概念に置き換えるステップと、

ａ）（ｖ）構文的に関連する前記ワード及び、より上位の前記概念のうち冗長となる部分を拡張後のクエリーから除去するステップとを有することを特徴とする検索方法。

【請求項１４】　請求項１３において、前記所定の基準は、前記ワードが用語辞書にあるかどうかに基づくことを特徴とする検索方法。

【請求項１５】　請求項１において、前記置き換えステップで、前記予備的インデックス内の、複数の意味を持つワードが、対応する、より上位の複数の概念によって置き換えられることを特徴とする検索方法。

【請求項１６】　請求項１２において、前記所定の基準を満たさないワードが固有名詞であることを特徴とする検索方法。

【請求項１７】　請求項１において、対応する、より上位の概念に関連する文書が、所定の数だけ検索されるまで、前記実行ステップが、連続する段階において続けられることを特徴とする検索方法。

【請求項１８】　請求項１７において、前記各段階が、１つの拡張クラスを表すことを特徴とする検索方法。

【請求項１９】　請求項１７において、前記各段階が、１つの拡張クラス内の１スロットを表すことを特徴とする検索方法。

【請求項２０】　請求項１７において、各段階で、文書が、少なくともクエリー内の１つのワードに割り当てられた重要度のレベルを反映した順序で検索されることを特徴とする検索方法。

【請求項２１】　サイズの小さい文書のインデックス、文書の含まれる元の細分度のワードに対応する、より上位の概念、及び前記インデックスと前記概念との間の関係を含む文書のデータベースを検索する方法であって、前記方法が、
ａ）元の細分度のクエリーのワードを、対応する、より上位の概念に置き換えることによって、文書のデータベースに適用されるクエリーを論理的に拡張するステップと、
ｂ）論理的に拡張された前記クエリーを実行し、前記インデックスを用いて、対応する、より上位の概念に関連する文書を検索するステップとを有することを特徴とする検索方法。
【請求項２２】　請求項２１において、論理的にクエリーを拡張する前記ステップが更に、
ａ）（ｉ）所定の基準を満たす、クエリーのワードのみを、より上位の意味的概念を有する、より上位の対応する概念に置き換えるステップを有することを特徴とする検索方法。
【請求項２３】　請求項２２において、より上位の意味的概念のそれぞれが、類義語を含むことを特徴とする検索方法。
【請求項２４】　請求項２１において、より上位の前記概念が、より上位の構文的概念であることを特徴とする検索方法。
【請求項２５】　請求項２４において、より上位の前記構文的概念のそれぞれが、あるレベルの頻度を越えて、文書内で共に発生するワードを含むことを特徴とする検索方法。
【請求項２６】　請求項２２において、論理的にクエリーを拡張する前記ステップが更に、
ａ）（ｉｉ）対応する、より上位の前記概念のそれぞれに対して、構文的に関連するワードを付加することによって、前記クエリーを更に論理的に拡張するステップと、
ａ）（ｉｉｉ）前記所定の基準を満たしていない、クエリー内のワードのそれぞれに対して、構文的に関連するワードを付加することによって、前記クエリーを更に論理的に拡張するステップとを有することを特徴とする検索方法。
【請求項２７】　請求項２６において、論理的にクエリーを拡張する前記ステップが更に、
ａ）（ｉｖ）所定の基準を満たす、構文的に関連する前記ワードを、関連する、より上位の概念に置き換えるステップと、
ａ）（ｖ）構文的に関連する前記ワード及び、より上位の前記概念のうち冗長となる部分を拡張後のクエリーから除去するステップとを有することを特徴とする検索方法。
【請求項２８】　請求項２７において、前記所定の基準は、前記ワードが用語辞書にあるかどうかに基づくことを特徴とする検索方法。
【請求項２９】　請求項２６において、前記所定の基準を満たさないワードが固有名詞であることを特徴とする検索方法。
【請求項３０】　請求項２６において、前記構文的概念のそれぞれが、あるレベルの頻度を越えて、文書内で共に発生するワードを含むことを特徴とする検索方法。
【請求項３１】　請求項２１において、
ｃ）関連性の順序に基づいて、検索された文書をランク付けするステップを、更に含むことを特徴とする検索方法。
【請求項３２】　請求項３１において、前記検索された文書が、元の細分度を有するクエリーのワードを用いてランク付けされることを特徴とする検索方法。
【請求項３３】　請求項３２において、関連性の順序は、クエリーのワードと検索された文書に含まれるワードが、厳密にマッチする場合を始めとし、以降、意味的にマッチする場合、構文的にマッチする場合、マッチしない場合の順であることを特徴とする検索方法。
【請求項３４】　請求項２１において、文書に含まれる元の細分度のワードが、複数の、より上位の概念に対応することを特徴とする検索方法。
【請求項３５】　請求項２１において、対応する、より上位の概念に関連する文書が、所定の数だけ検索されるまで、前記実行ステップが、連続する段階において続けられることを特徴とする検索方法。
【請求項３６】　請求項３５において、前記各段階は、１つの拡張クラスを表すことを特徴とする検索方法。
【請求項３７】　請求項３５において、前記各段階は、１つの拡張クラス内の１スロットを表すことを特徴とする検索方法。
【請求項３８】　請求項３５において、各段階で、文書が、少なくともクエリー内の１つのワードに割り当てられた重要度のレベルを反映した順序で検索されることを特徴とする検索方法。
【請求項３９】　文書の予備的インデックス、文書内に含まれるワード、及び前記インデックスと前記ワードとの間の関係を含み、前記インデックス内のワードが元の細分度である、文書のデータベースを検索するシステムであって、前記システムが、
ａ）より粗い細分度の小さなサイズのインデックスを生成するために、前記予備的インデックスの中のワードを、対応する、より上位の概念に置き換えるインデクサと、
ｂ）前記文書のデータベースに適用されるクエリーを提供するためのユーザ・インタフェースと、
ｃ）元の細分度を有する、クエリーのワードを、対応する、より上位の概念に置き換えることによって、前記クエリーを論理的に拡張し、論理的に拡張された前記クエ

リーを、より粗い細分度のインデックスを使用して実行し、対応する、より上位の概念に関連する文書を検索するプロセッサとを有することを特徴とする検索システム。

【請求項４０】　請求項３９において、前記プロセッサが、関連性の順に、検索された文書をランク付けすることを特徴とする検索システム。

【請求項４１】　請求項４０において、前記プロセッサが、元の細分度を有する、クエリーのワードを使用して、検索された文書をランク付けすることを特徴とする検索システム。

【請求項４２】　請求項４１において、関連性の順序は、クエリーのワードと検索された文書に含まれるワードが、厳密にマッチする場合を始めとし、以降、意味的にマッチする場合、構文的にマッチする場合、マッチしない場合の順であることを特徴とする検索システム。

【請求項４３】　請求項３９において、より上位の前記概念は、より上位の意味的概念であることを特徴とする検索システム。

【請求項４４】　請求項４３において、より上位の前記意味的概念のそれぞれが、類義語を含むことを特徴とする検索システム。

【請求項４５】　請求項３９において、前記インデクサが、所定の基準を満たす予備的インデックス内のワードのみを、対応する、より上位の概念で置き換えることを特徴とする検索システム。

【請求項４６】　請求項４５において、前記所定の基準は、前記ワードが用語辞書にあるかどうかに基づいていることを特徴とする検索システム。

【請求項４７】　請求項３９において、より上位の前記概念が、より上位の構文的概念であることを特徴とする検索システム。

【請求項４８】　請求項４７において、より上位の前記構文的概念のそれぞれが、あるレベルの頻度を越えて文書内に共に発生するワードを含むことを特徴とする検索システム。

【請求項４９】　請求項３９において、前記プロセッサが更に、

ｃ）（ｉ）所定の基準を満たす、クエリーのワードのみを、より上位の意味的概念である、対応する、より上位の概念に置き換えることによって、論理的にクエリーを拡張することを特徴とする検索システム。

【請求項５０】　請求項４９において、前記プロセッサが更に、

ｃ）（ｉｉ）対応する、より上位の前記概念のそれぞれに対して、構文的に関連するワードを付加し、

ｃ）（ｉｉｉ）前記所定の基準を満たしていない、クエリー内のワードのそれぞれに対して、構文的に関連するワードを付加することによって、前記クエリーを論理的に拡張することを特徴とする検索システム。

【請求項５１】　請求項５０において、前記プロセッサが更に、

ｃ）（ｉｖ）所定の基準を満たす、構文的に関連する前記ワードを、関連する、より上位の概念に置き換え、

ｃ）（ｖ）構文的に関連する前記ワード及び、より上位の前記概念のうち冗長となる部分を拡張後のクエリーから除去することによって、前記クエリーを論理的に拡張することを特徴とする検索システム。

【請求項５２】　請求項５１において、前記所定の基準は、前記ワードが用語辞書にあるかどうかに基づいていることを特徴とする検索システム。

【請求項５３】　請求項３９において、複数の意味を有する、前記予備的インデックス内のワードが、対応する、より上位の複数の概念に置き換えられることを特徴とする検索システム。

【請求項５４】　請求項５０において、前記所定の基準を満たさないワードが固有名詞であることを特徴とする検索システム。

【請求項５５】　請求項３９において、対応する、より上位の概念に関連する文書が、所定の数だけ検索されるまで、前記クエリーの実行が、連続する段階において続けられることを特徴とする検索システム。

【請求項５６】　請求項５５において、前記各段階が、１つの拡張クラスを表していることを特徴とする検索システム。

【請求項５７】　請求項５５において、前記各段階は、１つの拡張クラス内の１スロットを表すことを特徴とする検索システム。

【請求項５８】　請求項５５において、各段階で、文書が、少なくともクエリー内の１つのワードに割り当てられた重要度のレベルを反映した順序で検索されることを特徴とする検索システム。

【請求項５９】　サイズの小さい文書のインデックス、文書の含まれる元の細分度のワードに対応する、より上位の概念、及び前記インデックスと前記概念との間の関係を含む文書のデータベースを検索するシステムであって、前記システムが、

ａ）前記文書のデータベースに適用されるクエリーを提供するためのユーザ・インタフェースと、

ｂ）元の細分度を有する、クエリーのワードを、対応する、より上位の概念に置き換えることによって、前記クエリーを論理的に拡張し、論理的に拡張された前記クエリーを、前記インデックスを使用して実行し、対応する、より上位の概念に関連する文書を検索するプロセッサとを有することを特徴とする検索システム。

【請求項６０】　請求項５９において、前記プロセッサが更に、

ｂ）（ｉ）所定の基準を満たす、クエリーのワードのみを、より上位の意味的概念を有する、より上位の対応する概念に置き換えることによって、前記クエリーを論理

的に拡張することを特徴とする検索システム。
【請求項６１】　請求項６０において、より上位の意味的概念のそれぞれが、類義語を含むことを特徴とする検索システム。
【請求項６２】　請求項５９において、より上位の前記概念が、より上位の構文的概念であることを特徴とする検索システム。
【請求項６３】　請求項６２において、より上位の前記構文的概念のそれぞれが、あるレベルの頻度を越えて、文書内で共に発生するワードを含むことを特徴とする検索システム。
【請求項６４】　請求項６０において、前記プロセッサが更に、
ｂ）（ｉｉ）対応する、より上位の前記概念のそれぞれに対して、構文的に関連するワードを付加し、
ｂ）（ｉｉｉ）前記所定の基準を満たしていない、クエリー内のワードのそれぞれに対して、構文的に関連するワードを付加することによって、前記クエリーを論理的に拡張することを特徴とする検索システム。
【請求項６５】　請求項６４において、前記プロセッサが更に、
ｂ）（ｉｖ）所定の基準を満たす、構文的に関連する前記ワードを、関連する、より上位の概念に置き換え、
ｂ）（ｖ）構文的に関連する前記ワード及び、より上位の前記概念のうち冗長となる部分を拡張後のクエリーから除去することによって、前記クエリーを論理的に拡張することを特徴とする検索システム。
【請求項６６】　請求項６５において、前記所定の基準が、前記ワードが用語辞書にあるかどうかに基づくことを特徴とする検索システム。
【請求項６７】　請求項６４において、前記所定の基準を満たさないワードが固有名詞であることを特徴とする検索システム。
【請求項６８】　請求項６４において、前記構文的概念のそれぞれが、あるレベルの頻度を越えて、文書内で共に発生するワードを含むことを特徴とする検索システム。
【請求項６９】　請求項５９において、前記プロセッサが更に、関連性の順序に基づいて、検索された文書をランク付けすることを特徴とする検索システム。
【請求項７０】　請求項６９において、前記検索された文書が、元の細分度を有するクエリーのワードを用いてランク付けされることを特徴とする検索システム。
【請求項７１】　請求項７０において、関連性の順序は、クエリーのワードと検索された文書に含まれるワードが、厳密にマッチする場合を始めとし、以降、意味的にマッチする場合、構文的にマッチする場合、マッチしない場合の順であることを特徴とする検索システム。
【請求項７２】　請求項５９において、文書に含まれる元の細分度のワードが、複数の、より上位の概念に対応することを特徴とする検索システム。
【請求項７３】　請求項５９において、対応する、より上位の概念に関連する文書が、所定の数だけ検索されるまで、前記クエリーの実行が、連続する段階において続けられることを特徴とする検索システム。
【請求項７４】　請求項７３において、前記各段階は、１つの拡張クラスを表すことを特徴とする検索システム。
【請求項７５】　請求項７３において、前記各段階は、１つの拡張クラス内の１スロットを表すことを特徴とする検索システム。
【請求項７６】　請求項７３において、各段階で、文書が、少なくともクエリー内の１つのワードに割り当てられた重要度のレベルを反映した順序で検索されることを特徴とする検索システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、一般的に、データベース内の文書を収集するのに適用されるインデックスとクエリーの分野に関する。より詳しくは、クエリーの効果的な拡張と処理、クエリーの拡張を実施するのに使用されるインデックスのサイズの縮小、及び連続的なクエリーの処理に関する。
【０００２】
【従来の技術】クエリーを適用することによって文書を検索する従来の検索システムは、文書を分類する共通の原理と方法論に基づいている。文書は通常、熟練者又は司書により、事前に指定され、調整された用語を用いて、手作業でインデックス付けされる。文書はまた、その文書に含まれる語（ワード）に基づいてインデックス付けされることもある。ユーザは、指定可能な用語から選択したワードと、それらの間を適当なブーリアン演算子で連結して文書の検索を行う。このようなタイプのシステムでは、厳密なマッチング戦略が用いられる。このアプローチは、単純で高精度といった多くの利点を有するものの、ワード・ミスマッチの問題が生じる。
【０００３】情報検索におけるワード・ミスマッチの問題は、作者がその文書で、ある概念を表すのに、あるワードを使用している場合に、ユーザが、それと同じ概念をクエリーにおいて指定する際、別のワードを使用してしまうことによって生じる。図１は、「ｃａｒ（乗用車）」及び「ｄｅａｌｅｒ（販売店）」に関連付けられた、ハイパーテキスト・マークアップ言語（ＨＴＭＬ）の文書において使用されるワードが、様々な文書の間で異なることがあることを示している。拡張可能なマークアップ言語（ＸＭＬ）や標準一般化マークアップ言語（ＳＧＭＬ）のような、ＨＴＭＬ以外の言語も用いられる。ユーザが、「ａｕｔｏｍｏｂｉｌｅ（自動車）」と「ｄｅａｌｅｒ（販売店）」というワードをクエリーに用いる場合、ワード・ミスマッチの問題で、対象となる

文書を１つも検索できない結果になる。

【０００４】尚、本明細書では、検索の対象が、主に英語を含むものと仮定しているため、検索に使用するクエリーの各要素は、英語で記述されている。しかし、これらは、ユーザの要求に応じて、どの国の言語で表現することも可能である。ここでは、前記英語で記述された要素に続いて（必要に応じ）括弧内に、その要素の日本語における意味を表すことにする。従って、当該括弧内の日本語は、単にクエリーの要素の意味を説明するためのものに過ぎず、クエリーの結果には影響を及ぼさない。

【０００５】クエリーの拡張は、このような問題を解決する技法として示唆されている。このアプローチは、意味の類似したワード（例えば、類義語や他の関連する意味を有するワード）及び構文的に関連するワード（例えば、一定の頻度以上で同じ文書内に同時に現れるワード群は、構文的共起ワードである）をクエリー内のワードとして用いることによってクエリーを拡張するものである。こうしてクエリーが拡張されると、関連する文書内のワードにマッチする可能性が高まる。クエリーの拡張が使用されると、「ｃａｒ　ｄｅａｌｅｒ（乗用車の販売店）」というワードを含むクエリーは、以下のように同様の意味の用語を含むように拡張される。

【０００６】行１．　［（「ｃａｒ（乗用車）」ＯＲ「ａｕｔｏｍｏｂｉｌｅ（自動車）」ＯＲ「ａｕｔｏ（車）」ＯＲ「ｓｅｄａｎ（セダン）」）ＯＲ
行２．　（「Ｆｏｒｄ（フォード車）」ＯＲ「Ｂｕｉｃｋ（ビュイック車）」）］ＡＮＤ
行３．　（「ｄｅａｌｅｒ（販売店）」ＯＲ「Ｓｈｏｗｒｏｏｍ（ショールーム）」ＯＲ「ＳａｌｅｓＯｆｆｉｃｅ（販売所）」）。

【０００７】上記例に含まれるクエリーの拡張には、２つのタイプがある。行１と行３のクエリーの拡張は、用語の意味において「ｃａｒ」と「ｄｅａｌｅｒ」に関連する追加ワードを追加するものである。即ち、意味的に類似するワードを追加するものである。「ａｕｔｏｍｏｂｉｌｅ」、「ａｕｔｏ」、及び「ｓｅｄａｎ」は、「ｃａｒ」というワードに類似する意味を有するワードである。同様に、「Ｓｈｏｗｒｏｏｍ」と「ＳａｌｅｓＯｆｆｉｃｅ」は、「ｄｅａｌｅｒ」というワードに類似する意味を有するワードである。他のタイプのクエリーの拡張は、行２に示すものであり、これは例えば、構文的共起関係によるものである。ワールドワイドウエブ（単にウエブとも言う）で用いられる多くのワードは、実際には固有名詞であり、用語辞書には見つからない。例えば、固有名詞は、Ｆｏｒｄ、Ｂｕｉｃｋ、ＮＢＡ、及びＮＦＬ（National Football League）といったものである。前述したように、構文的共起関係は、２つのワードが、同じ文書に同時に現れる頻度を分析することによって導出される。これは、２つのワードが頻繁に同じ文書内に現れる場合には、それらのワードが関連している可能性が高いという仮定に基づくものである。例えば、「Ｆｏｒｄ」と共に発生するワードとして、「ｄｅａｌｅｒ（販売店）」、「ｂｏｄｙ　ｓｈｏｐ（車体工場」、「Ｍｕｓｔａｎｇ（マスタング：フォード社製の車の名前）」、「Ｅｓｃｏｒｔ（エスコート：フォード社製の車の名前）」等が考えられる。

【０００８】クエリーの拡張を支援するために、用語の意味によって関連付けられたワードのインデックスと、共起情報のような構文的関係が適切に維持されなければならない。用語の意味によってワードに関連付けられたインデックスは、階層構造、意味ネットワーク、又は関連ワードの階層クラスタとして構成される。前記階層構造については、１９９７年８月、ギリシャのアテネで行われた、the 23rd International Conference on Very Large Data Basesの予稿集のページ538-547、W. Li他の「Facilitating Multimedia Database Exploration through Visual Interfaces and Perpetual Query Reformulations」を参照されたい。また、前記意味ネットワークについては、１９９０年、International Journal of Lexicography 3(4)、ページ245-264における、G. A. Millerの「Nouns in WordNet: A Lexical Inheritance System」を参照のこと。また、関連ワードの階層クラスタについては、１９８３年、ニューヨーク、McGraw-Hill、ページ118-155の、G. Salton他による「The SMART and SIRE Experimental Retrieval Systems」を参照のこと。構文的共起関係のような構文的関係は、２項関係で表されるので、構文的関係のインデックスのサイズは非常に大きい。この問題を解決するため、いくつかの技法が提案されている。これらの技法については、１９９２年、デンマークにおけるthe Fifteenth annual International ACM SIGIR Conferenceの予稿集の、G. Grefenstetteによる「Use of syntactic context to produce term association lists for text retrieval」、１９９６年、スイスのチューリッヒにおけるthe 19th Annual International ACM SIGIR Conferenceの予稿集の、J. Xu他による「Query Expantion Using Local and Global Document Analysis」、１９９７年、アメリカ合衆国ペンシルバニア州フィラデルフィアにおける、the20th Annual International ACM SIGIR Conferenceの予稿集の、C. Jacqueminによる「Guessing Morphology from Terms and Corpora」を参照のこと。こうした技法は、発生頻度の分析、及び形態素規則（例えば、全てのワードをその起源となる形態に変換する）や用語辞書の使用を含むものである。

【０００９】ワード・ミスマッチの問題に関しては、情報検索（ＩＲ）の分野において、かなりの研究がされてきている。これについては、１９８３年、McGraw-Hill BookCompany発行の、G. Salton他による「Introduction to Modern Information Retrieval」、１９８９年、Addison-Wesley Publishing Company, Inc発行の、G.Salt

onによる「Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer」、及び１９９７年、アメリカ合衆国カリフォルニア州サンフランシスコ、Morgan Kaufmannの、K. Sparck Jones他による「Readings in Information Retrieval」を参照のこと。

【００１０】しかし、この研究の殆どが、適合率と再現率といった、検索の基準に関する点を指向したものである。クエリーの拡張を効果的に支援する方法（１９９３年、メリーランド州Gaithersburgで行われたthe 3rd Text Retrieval Conferenceの予稿集の、C. Buckley他による「Automatic Query Expansion Using SMART」参照）やインデックス付けのメカニズムを示唆した研究がいくつか有るが、満足する解決法のない問題が依然として２つ残っている。第１の問題は、ある文書の集合（例えば、ウエブ）内の多くのワードが別個の固有名詞であり、各ワードが意味的に同じワード及び構文的に関連したワードを多く有するので、インデックスのサイズが極めて大きくなってしまうことである。第２の問題は、クエリーが追加ワードによって拡張されるので、クエリーの処理コストが高くなってしまうことである。

【００１１】ウエブから収集された文書情報を取り扱う際には、文書の数が非常に多くなり、使用されているワードが極めて多様で、一貫性がなく、時には間違っている（例えば、タイプエラー）ため、これらの問題は、ますます顕著になる。ある研究では、ウエブに関する殆どのユーザ・クエリーは、通常、ワードを２つ有している。これについては、１９９５年、Digital Libraries (DL '95)の予稿集で、B.Croft他の「Providing Government Information on the Internet: Experienceswith THOMAS」を参照されたい。しかし、クエリー拡張を用いれば、クエリーの長さは実質的に長くなる。結果的に、ウエブ上の既存のサーチエンジンのほとんどは、クエリー拡張機能を提供できないことになる。

【００１２】ここで、クエリー拡張の分野における既存の研究を概説する。クエリー拡張は、ＩＲの分野において、かなりの注目を集めた。しかし、いままで注目されてきた部分は、クエリーの拡張によって、改善される検索の基準（即ち、適合率及び再現率）の程度を評価することであった。別の研究では、与えられたクエリーのワードに関して、１組の類似する用語を識別するために、辞書を構築することに焦点があてられてきた。しかし、今までの研究は、クエリーが拡張された場合のクエリーの効率的な処理の問題や、クエリーの拡張及び処理を行うのに用いられるインデックスのサイズを小さくするといった点に取り組んでいない。更に、厳密なマッチング及び類似的なマッチングに基づいて文書をランク付けする問題は、困難なものとして残されたままである。

【００１３】ＳＭＡＲＴは、よく知られた先進の情報検索システムの１つである。これに関しては、１９７１年、アメリカ合衆国ニュージャージー州Englewood CliffsのPrentice-Hallから発行されたGerard Salton編集のThe SMART Retrieval System -Experiments in Automatic Document Processing、第１２章の、R. T. Dattolaによる「Experiments with a fast algorithm for automatic classification」、及び上記文献の、G. Salton他による「The SMART and SIRE Experimental Retrieval Systems」を参照のこと。ＳＭＡＲＴでは、各文書が用語のベクトルで表される。ベクトルのそれぞれの位置は、文書内の対応する用語の重み（重要性）を表している。Ｎ個の異なる用語を有するＭ個の文書の集合は、Ｍ×Ｎの行列で表される。クエリーもまた用語のベクトルとして表される。文書の検索は、クエリー・ベクトルと各文書のベクトルとの余弦に対応する類似性の計算に基づく。他の、よく知られたシステムには、ＩＮＱＵＥＲＹがある。これについては、１９９５年、Information Processing and Managementの3:327-332で、J. Callan他による「Trec and tipster experiments with inquery」を参照のこと。

【００１４】潜在的意味インデックス（ＬＳＩ）は、辞書的なマッチングによる個別の用語検索の替わりに、統計的に導出された概念インデックスに依存する技法である。これについては、１９９０年、Journal of the America Society of Information Science、41:391-407の、R. Harshman他による「Indexing by latent semantic analysis」、及び１９９５年、the 1995 ACM Conference on Supercomputingの予稿集で、M. W. Berry他による「Computational Method for Intelligent Information Access」を参照されたい。ＬＳＩは、ワードの使用法に、いくつかの見えない構造、即ち潜在的な構造があることを仮定し、その構造は、文書におけるワードの発生を分析することによって外部化される必要がある。従って文書は、非常に大きな範囲の用語空間におけるベクトルとして考えられ、そのベクトルの個々の要素は与えられた文書における特定の用語の発生頻度を表している。全体及び局所的重み付けに基づく、より洗練された基準も使用されうる。短縮された特異値分解（ＳＶＤ）が、文書に亘るワード使用の構造を評価する。これについては、１９８９年、アメリカ合衆国メリーランド州ボルチモアのJohns-Hopkinsの、G. Golub他による「Matrix Computations」第２版を参照されたい。ここでは、検索が、特異値を有するデータベース、及び短縮されたＳＶＤから得られたベクトルを使用して実行される。ＬＳＩの予備的評価では、この情報検索のアプローチは、個々の用語に基づくものより粗い基準とされている。

【００１５】自動化されたクエリー拡張は、ワード・ミスマッチ問題を取り扱う技法として長い間示唆されてきた。これについては、１９９４年、アイルランド共和国ダブリンで行われたthe 17th Annual International ACM SIGIR Conferenceの予稿集で、E. Voorheesによる「Q

uery Expansion Using Lexical-Semantic Relations」を参照されたい。あるアプローチでは、類語辞典を用いてクエリーを拡張し、関連する文書内でワードがマッチする可能性を高めている。研究では、単に一般的な類語辞典を用いるだけでは、改善に限界があることが分かっている。多くの革新的技法も提案されている。１９９４年、the 3rd International Conferenceon Information and Knowledge Managementの予稿集の、O. Kwon他による「Query Expansion Using Domain Adapted, Weighted Thesaurus in an Extended Boolean Model」、１９９３年、アメリカ合衆国ペンシルバニア州ピッツバーグで行われたthe 16th Annual International ACM SIGIR Conferenceの予稿集の、E. Voorheesによる「Concept Based Query Expansion」、同予稿集の、E. Voorheesによる「Query Expansion Using Lexical-Semantic Relations」、及び同予稿集の、M. W. Berry他による「Computational Methods for Intelligent Information Access」を参照されたい。実験の結果、自動化されたクエリー拡張では、平均で７％から２５％の検索の効率化がはかられている。これについては、同予稿集の、C. Buckley他による「Automatic Query Expansion Using SMART」を参照されたい。

【００１６】クエリーの改良は、構文的に関連するワードを含めることによっても達成される。このアプローチは、ワードを、文書内での共起情報に基づいてクラスタ化し、これらのクラスタを用いてクエリーを拡張する。この共起情報は、２項関係であるため、こうしたインデックスのサイズは常に、極めて大きなものになる。また、あるグループは、ワードの変形に関する共起統計の集大成を用いてステマ（stemmer）を変更又は生成し、形態素規則のみを用いたアプローチに較べてどれだけ有利かを実証した。これについては、１９９４年、the Fourth Annual Symposiumの予稿集の、W. B. Croft他の「Corpus-Specific stemming Using Word Form Co-occurrence」を参照されたい。クエリーの用語を１組の意味的に関連する用語に拡張する上記各技法は、全体(global)分析と呼ばれる。クエリー拡張では、関連フィードバックからの用語もクエリーに追加され、検索の効率を改善する。１９９０年６月、Journal of the American Society for Information Scienceの41(4):288-297、G. Salton他の「Improving retrieval performance by relevance feedback」を参照のこと。これは、局所(local)分析と呼ばれる。これまでの研究では、ワードの前後関係及び語句の構造を用いた全体分析技法を文書の一部分の組に適用することによって、単純な局部的フィードバックより効果的でより確実な検索結果が得られることを示している。詳細については、上記文献の、J. Xu他による「Query Expansion Using Local and Global Document Analysis」を参照のこと。

【００１７】しかし、前述したように、いままでの研究は、クエリーが拡張された場合のクエリーの効率的な処理の問題を解決したり、クエリー拡張とクエリー処理を実行するのに用いられるインデックスのサイズを小さくすることを目指すものではなかった。

【００１８】

【発明が解決しようとする課題】本発明の目的は、ワード・ミスマッチの問題と、結果的に生じるクエリー処理の非効率さを解決するために、小さなサイズのインデックスを使用して効率的なクエリー拡張を行い、連続的なクエリーの処理を行う方法及び装置を提供することである。より詳しくは、クエリー内に指定されたワードと意味的に類似し、構文的に関連のあるワードを用いて、そのクエリーを、物理的ではなく概念的に拡張し、結果的に関連する文書を逃すことを少なくする。

【００１９】また、クエリーの拡張を支援するために、用語の意味について関連するワード、及び構文的共起関係にあるワードのインデックスが維持される必要があり、こうしたクエリー拡張の支援に関しては、以下の２つの問題が重要になる。１つ目はインデックス・テーブルのサイズの問題であり、２つ目はクエリー処理のオーバーヘッドの問題である。本発明は、これらの問題を解決することも目的とする。

【００２０】

【課題を解決するための手段】本発明によれば、複数の細分度からなる情報の概念と処理構造が、クエリーの拡張を支援するために使用される。本発明は、インデックス付けフェーズ、クエリー処理フェーズ、及びランク付けフェーズを含む。インデックス付けフェーズでは、意味的に類似したワードが１つの概念としてグループ化され、こうして、より粗く細分化された意味概念のために、結果的に実際の１つのインデックス・サイズが小さくなる。クエリー処理の間、クエリー内のワードが、辞書と実際のデータの内容を使用して、対応する意味概念及び構文的拡張にマッピングされ、結果的に元のクエリーに対して論理的な拡張が行われる。更に、処理に関するオーバーヘッドが回避される。次に、最初のクエリーのワードは、検索結果として得られた文書を、厳密なマッチング、意味的なマッチング、及び構文的マッチングに基づいてランク付けするのに用いられ、連続的なクエリーの処理を実行するのにも用いられる。

【００２１】

【発明の実施の形態】本発明による、効率的にクエリーの拡張を行うための方法及び装置の好適実施形態が、添付図面と共に以下で詳細に説明される。以下の説明は、ＮＥＣのＰＥＲＩＣＯオブジェクト指向データベース管理システム（ＯＯＤＢＭＳ）に関してなされるが、本発明はこれに限られるものではないことに注意すべきである。本発明は、様々なデータベース・システム及び文書の集合体に適用されうる。

【００２２】本発明は、複数の細分度の概念を導入する

ことによって、クエリーの拡張に関して、効果的なインデックス付けと処理支援を提供する。本発明のアプローチは、ワードのステミング（stemming）の後で、利用可能な技法を用いて、意味的に類似するワードと構文的に関連するワードについて、インデックスを設定する。前記技法については、１９９６年、スイス、チューリッヒでのthe 19th AnnualInternational ACM SIGIR Conferenceの予稿集の、J. Xu他の「Query ExpansionUsing Local and Global Document Analysis」、及び１９９７年、アメリカ合衆国ペンシルバニア州フィラデルフィアでのthe 20th Annual International ACM SIGIR Conferenceの予稿集の、C. Jacqueminの「Guessing Morphology from Terms and Corpora」を参照のこと。更に本発明のアプローチは、いくつかのエントリ（タプル）を、より高レベルの細分度で１つのエントリにマージすることにより、インデックスのサイズを小さくする。クエリー処理の間、より高いレベルの細分度での情報を有した、そのタプルが、関連文書を検索するのに用いられる。その後、クエリーの元のワードは、より細かい細分度で、厳密なマッチング、意味的に類似するマッチング、及び構文的に関連するマッチングに基づいてクエリー処理の間に結果として得られる文書をランク付けするために用いられる。複数の細分度を有するインデックスとクエリー処理技法を使用することによって、検索メカニズムにおける全体の精度を維持したまま、インデックスのサイズを小さくすることができ、かつ、より速いクエリー処理を実現できる。

【００２３】最初に、複数の細分度の表記と、それが、どのように、ほとんどのＩＲシステムによって使用されている従来のインデックス付けに関連して適応されるのかについて説明する。次に、所定の文書の集合に関して、複数の細分度を有するインデックス付けを行う場合の、記憶域に対するオーバーヘッドについての見積りを行う。

【００２４】従来のＩＲシステムは、文書リストから所与のワードを容易に検索するために、インデックスを保持し、同時に、得られた文書に関連付けられたワードの組を抽出する。この場合、「文書」という用語は、テキスト、イメージ、又はテキストとイメージの組み合わせに関連することに注意すべきである。

【００２５】図２は、インデックスの例を示している。図２の（ｂ）に示すテーブルは、図２の（ａ）に示すテーブルを転置したインデックスである。図２では、説明を容易にするため、これらのインデックスがテーブルの形で示されている。しかし、実際の環境では、例えばＮＥＣのＰＥＲＣＩＯ　ＯＯＤＢＭＳの上位層のクラスが用いられる。１つのクエリーの例をとると、ユーザが最初に、ワード「ｃａｒ（乗用車）」かつ「ｄｅａｌｅｒ（販売店）」を用いてクエリーを作成すると、ＩＲシステムは、図２の（ｂ）のテーブルの対応する行から文書リストを取り出す。この場合、クエリーの解答は、２つの行から得られた文書リストの共通部分となる。このＩＲに対するアプローチは、明らかに、厳密なマッチングのみを支援するものであり、「ａｕｔｏｍｏｂｉｌｅ　ｄｅａｌｅｒ（自動車の販売店）」、「ｃａｒ　ｓｈｏｗｒｏｏｍ（乗用車のショールーム）」、又は「ａｕｔｏｍｏｂｉｌｅ　ｓｈｏｗｒｏｏｍ（自動車のショールーム）」といった類似の意味を有する用語を含む関連文書を得ることができない。クエリー拡張は、クエリーを「ｃａｒ」かつ「ｄｅａｌｅｒ」という記述から、（「ｃａｒ」又は「ａｕｔｏｍｏｂｉｌｅ」）かつ（「ｄｅａｌｅｒ」又は「ｓｈｏｗｒｏｏｍ」）という記述に拡張する特別のユーティリティと関連して使用される。このアプローチは実現可能ではあるが、クエリー処理にかなりのオーバーヘッドを招くことになる。特に、図２の（ｂ）のインデックス・テーブルについての２回のルックアップの代わりに、元のクエリー内のワードと意味的に類似するワードのそれぞれについて、何回かのルックアップが必要になる。また、オンライン辞書のような類語辞書的ツールが、クエリーの用語を、それらと意味的に類似する用語に拡張するのに必要である。これらの観察から、本発明は、文書の集合を検索する際に、クエリーの拡張を支援する、より効果的な方法を提供する。

【００２６】先に述べたように、ユーザの語彙と作者の語彙とのミスマッチを避けるために、意味の類似するワード、及び構文的関係を有するワードを用いてクエリーを拡張する方法に基づいたクエリーの拡張が必要とされる。

【００２７】図３は、従来のＩＲシステムにおいて、クエリーの拡張を容易にするのに追加が必要となるデータ構造を示している。特に、図３は、各ワードが意味的に類似する概念にグループ化される、用語の意味を含むオンライン辞書から導出されたテーブルを示している。なお、図３に示されたテーブルは、説明のため簡略化されている。例えば、類似する用語の組「ｃａｒ（乗用車）」、「ａｕｔｏ（車）」、「ａｕｔｏｍｏｂｉｌｅ（自動車）」、及び「ｓｅｄａｎ（セダン）」は、１つの象徴的エンティティ、ｓｅｍ１として表されている。辞書や類語辞書に基づく意味的な類似とは違って、ＩＲにおける構文的関係は、文書の収集そのものによって決定される。特に、ワードの共起情報は、２つのワードを構文的に関連付けるのに使用される。図３（ｂ）は、この情報を表したインデックスを例示している。図３の補助インデックスと共に、図２の従来のＩＲインデックスを用いることによって、基本的なクエリー拡張技法が、ＩＲシステムにおいて支援される。基本的には、ユーザのクエリーが与えられると、クエリーのワード・リストが、意味的に類似するワード及び構文的に関連するワードを含むように拡張される。

【００２８】クエリーの拡張を用いたクエリーの処理には、上述の方法が使用されるが、このアプローチでは、処理に関するオーバーヘッドが高くなってしまう。本発明によれば、クエリーをより効率的に処理することができる追加のインデックス構造が使用される。本発明のアプローチの基本的発想は、図２及び図３のインデックスを、クエリーが概念的に拡張されるように変換するものである。即ち、意味的に類似するワード及び構文的に関連するワードをリスト内に含ませることによって、クエリーのワードのリストを物理的に拡張するのではなく、クエリーのワードを、その関連する、より上位レベルの意味概念と構文的関係（例えば、共起関係）のワードと入れ替えることによって、クエリーを概念的に拡張する。このことは、追加のインデックス構造による容量オーバーヘッドの追加をもたらす。しかし、ユーザのクエリーがより効率的に処理されるので、全体としては節約を達成できる。

【００２９】前述したように拡張されたクエリーを処理するために、図4に示すように、インデックス・テーブルが変更される。特に、図4の（ａ）に示すインデックス・テーブルは、各ワード（固有の名称でない）を、より上位レベルの意味概念のワードに置き換えることによって、図2の（ａ）から導出される。図4の（ｂ）に示すインデックス・テーブルは、図2の（ｂ）に示されたワードを、それらが対応する、より上位レベルの意味概念のワードと組み合わせ、それぞれの文書リストのエントリをマージすることによって得られる。従って、「ｃａｒ」、「ａｕｔｏ」、「ａｕｔｏｍｏｂｉｌｅ」、及び「ｓｅｄａｎ」に対応する行エントリは、図4の（ｂ）では単一のエントリＳｅｍ１として表されている。同様に、図２（ｂ）の、「ｄｅａｌｅｒ」、「ｓｈｏｗｒｏｏｍ」、及び「ＳａｌｅｓＯｆｆｉｃｅ」に対応する行は、Ｓｅｍ２というラベルの１行に纏められている。

【００３０】構文的に関連するワードに対するインデックスは通常、いくつかの理由から、意味的に関連するワードに対するインデックスよりかなり大きい。ウエブ上の多くのワードは、固有の名称であり、辞書には見つからない。実験では、2,904の文書を分析した場合、キーワードの４２％だけがＷｏｒｄＮｅｔで見つかった。ＷｏｒｄＮｅｔは60,000以上のワードを有するオンライン辞書である。これについては、１９９０年、International Journal of Lexicography 3(4)、ページ245-264の、G. A. Millerによる「Nouns in WordNet: A Lexical Inheritance System」を参照されたい。残りの５８％のワードは固有名詞やタイプエラーを含んでおり、これがインデックスのサイズを肥大化させる元となっている。従来のＩＲシステムにおいては、構文的な関連付けは、通常、共起関係によって把握されていた。同じ文書内でのワードの共起関係は、１対１関係であるため、ｎ個のワードが識別された場合、インデックスのサイズは、最悪のケースでは、（ｎ×（ｎ−１））／２となる。巨大な記憶域とインデックス付けのオーバーヘッドのために、３個以上のワードの共起関係をインデックス付けするのは、非常にコストがかかる。

【００３１】辞書に見つかったワード（意味的に意義のあるもの）をＳとし、他の全てのワード（固有名詞）をＰとする。辞書にあるワードと辞書にないワードという、上記分類に基づいて、ワードの間の共起関係が３つの異なるカテゴリに分類される。

【００３２】・Ｐ−Ｐ型：例えば（Ｔｏｙｏｔａ（トヨタ）、Ａｖａｌｏｎ（トヨタ車の名前））、（Ａｃｕｒａ（アキュラ）、Ｌｅｇｅｎｄ（アキュラ車の名前））、（Ｎｉｓｓａｎ（日産）、Ｍａｘｉｍａ（日産車の名前））。

【００３３】・Ｓ−Ｐ型、又はＰ−Ｓ型：例えば（Ｂｕｉｃｋ（フォード車の名前）、ｃａｒ（乗用車））、Ｂｕｉｃｋ、ｄｅａｌｅｒ（販売店））、（ｃａｒ、Ｆｏｒｄ（フォード社））、（Ｆｏｒｄ、ａｕｔｏ（車））、（Ｆｏｒｄ、ｄｅａｌｅｒ）。

【００３４】・Ｓ−Ｓ型：例えば（ｃａｒ、ｇａｒａｇｅ（ガレージ））、（ａｕｔｏ、ｇａｒａｇｅ）。

【００３５】通常、図3の（ｂ）に示す、より粗い細分度に変換できないＰ−Ｐ型のエントリを変換することは困難である。しかし、他の全てのエントリは、対応する、より高いレベルの意味概念に置換できるＳワードを有する。これによって、共起インデックスのサイズが減少し、クエリー処理のスピードアップが実現される。インデックスのサイズの減少は、以下のように生じる。Ｓ−Ｐ型（w_i、Ｘ）の各エントリに対し、w_iが意味概念Sem_iに対応するように、図3の（ｂ）に示された全ての（w_i、Ｘ）のエントリを、図4の（ｃ）の（Sem_i、Ｘ）に置換する。ここで、対応する文書のリストもマージされる。同様の手順がＰ−Ｓ型のエントリにも適用される。図4の（ｃ）に示すように、エントリ（Ｆｏｒｄ、ｃａｒ）と（Ｆｏｒｄ、ａｕｔｏ）は、（Ｆｏｒｄ、Ｓｅｍ１）に置換される。同様に、エントリ（Ｆｏｒｄ、ｄｅａｌｅｒ）と（Ｆｏｒｄ、ｓｈｏｗｒｏｏｍ）は、（Ｆｏｒｄ、Ｓｅｍ２）に置換される。こうしたマージ・メカニズムについて、図5の（ａ）（ｂ）を用いて説明する。

【００３６】Ｓ−Ｓ型のエントリは、以下の２つの方法でマージされる。

【００３７】・単一マージ：図5の（ａ）（ｂ）に示すような、１対多／多対１のタイプのマージ。例えば、エントリ（ｃａｒ、ｄｅａｌｅｒ）、（ａｕｔｏｍｏｂｉｌｅ、ｄｅａｌｅｒ）、及び（ａｕｔｏ、ｄｅａｌｅｒ）は、（Ｓｅｍ１、ｄｅａｌｅｒ）に置換される。ここで使用されるアルゴリズムは、Ｓ−Ｐ型及びＰ−Ｓ型で使用されるものと同じである。

【0038】・複合マージ：図5の（ｃ）に示すような、多対多のタイプのマージ。例えば、エントリ（ｃａｒ、ｄｅａｌｅｒ）、（ａｕｔｏｍｏｂｉｌｅ、ｓｈｏｗｒｏｏｍ）、及び（ａｕｔｏ、ＳａｌｅｓＯｆｆｉｃｅ）は、（Ｓｅｍ１、Ｓｅｍ２）に置換される。このタイプのマージのアルゴリズムは、以下のようなものである。

【0039】１．Ｓ－Ｓ型の各エントリ（w_i、Ｘ）に対して、w_iが、意味概念Sem_iに対応するように、図3の（ｂ）の（w_i、Ｘ）のエントリ全てを、図4の（ｃ）に示すような（Sem_i、Ｘ）に置換する。

【0040】２．（Sem_i、w_j）のタイプの各エントリに対して、w_jが、意味概念Sem_jに対応するように、こうした全ての（Sem_i、w_j）を、（Sem_i、Sem_j）に置換する。

【0041】上記ステップ２は、上記ステップ１の前に実行することもできることに注意すべきである。更に、このアルゴリズムのステップ１とステップ２は、マージするものがなくなるまで繰り返し行われうる。

【0042】複数のエントリがマージされると、それに応じて、各エントリの構文的ワードリストも、合併（ＵＮＩＯＮ）演算によってマージされる。

【0043】複数の細分度を有するインデックス付け技法は、ＯＯＤＢＭＳの上位層に実装されうる。こうした実装では、図2の（ａ）、図3の（ａ）、及び図4の（ｃ）に示すテーブルは、内容を有するクラスである。他のテーブルは、ポインタのみを有するクラスである。インデックスに対する更新、削除、及び挿入操作は、自動監視維持やクラスの間で伝達を行うプログラムを介して、ＯＯＤＢＭＳによって実行される。複数の細分度を有するインデックスの維持は累積的に行われ、再編成は必要とされない。

【0044】次に、従来のワードに基づくインデックスの他に、意味概念に基づくインデックス・テーブルを支援するのに必要なために追加される記憶域のオーバーヘッドを考慮に入れて、本発明による実施例の見積りを計算する。前述したように、図4に示すテーブルが、効率的なクエリー処理のために導入される。最初に、従来のＩＲシステムで使用されるインデックス、即ち、図2に示すテーブルに関する記憶域の見積りに関する計算を行う。所定の集合体における文書の数はＤであるとする。更に、その所定の文書の集合体における、辞書にあるワード数（ワード・ステミングを用いて、ストップ・ワードとグルーピング・ワードを取り除いた後の数）はＷであり、辞書にないワードの数はＶとする。また、文書毎の、辞書にあるワード数の平均をｗとし、辞書にないワード数の平均をｖとし、ワード毎の文書数の平均をｄとする。エントリ数（即ち、行数）とテーブルの全体のサイズ（即ち、ポインタの数）に基づいて、インデックスのサイズが計算される。テーブルの各エントリが、ポインタ・データとして表されることに注意すべきである。これらのパラメータが与えられると、図2の（ａ）に示されたテーブルのサイズは、以下の式（２）で表される。

【0045】

行数〔2(a)〕　＝Ｄ　...（１）

全体サイズ〔2(a)〕＝（１＋ｖ＋ｗ）Ｄ　...（２）。

【0046】各行において、文書の識別のために１つのポインタが必要とされ、ワードのリストの中で辞書にないワードを表すのに、平均でｖ個のポインタが必要とされ、更にワードのリストの中で辞書にあるワードを表すのに、平均でｗ個のポインタが必要とされるので、（１＋ｖ＋ｗ）の項が生じていることに注意すべきである。同様に、図2の（ｂ）に示すテーブルのサイズは、以下の式（４）で表される。

【0047】

行数〔2(b)〕　＝Ｗ＋Ｖ　...（３）

全体サイズ〔2(b)〕＝（１＋ｄ）・（Ｗ＋Ｖ）　...（４）。

【0048】このテーブルの各行は、平均で、文書リスト内で文書の識別子となるｄ個のポインタと、ワードそのものを指す１つのポインタを必要とする。

【0049】次に、基本的なクエリー拡張を支援するのに必要な、オンライン辞書と構文的共起テーブルの記憶域オーバーヘッドを見積る。ｆを、辞書にあるワードを意味概念にグループ化することによって得られる圧縮要素とする。従って、ｆは、１つの概念にグループ化されたワードの数の平均と見ることができる。図3の（ａ）に示すテーブルのサイズは、以下の式（６）のように表すことができる。

【0050】

行数〔3(a)〕　＝Ｗ／ｆ　...（５）

全体サイズ〔3(a)〕＝Ｗ＋Ｗ／ｆ　...（６）。

【0051】式（５）は、辞書にあるワードの記憶域が、圧縮要素ｆに基づいて圧縮されるので、このように表される。式（６）は、Ｗ個のポインタが、ワードのリスト内のワードを表すのに必要で、Ｗ／ｆ個のポインタが意味的な識別子を表すのに必要であることを示している。図3の（ｂ）で示されるテーブルのサイズは、最悪の場合、以下の式（８）で表される。

【0052】

行数〔3(b)〕　＝Ｖ（Ｖ－１）／２＋ＶＷ＋Ｗ（Ｗ－１）／２　...（

7)

全体サイズ[3(b)]＝（１＋２＋ｑ）・（Ｖ（Ｖ－１）／２＋ＶW+W（W－１）／２）　．．．（８）。

【００５３】式（７）では、第１項がＰ－Ｐ型のワードの共起関係に対応し、第２項がＳ－Ｐ型又はＰ－Ｓ型に対応し、最後の項はＳ－Ｓ型の共起関係に対応する。ｑは、共起関係を表す項毎の、文書リスト内のエントリの平均数を表している。更に、構文的用語識別子を表すために３つのポインタが必要とされ、２つのワードが各行の共起関係に含まれる。

【００５４】次に、意味的に類似する用語の組を１つのユニークな意味概念にグループ化する、本発明による複数の細分度を有するインデックス付けに関する記憶域オーバーヘッドについて見積りを行う。前述したように、図４に示すインデックス・テーブルのサイズを計算するために、意味概念毎の平均文書数と、文書毎の意味概念の平均数を見積る必要がある。複数の用語が１つの意味概念に縮減されるので、意味概念毎の文書の平均数は、ｄより大きくなり、この拡張は、ｆ・ｄにはならないことを示すことができる。他方、文書毎の概念の平均数は、ｗを越えることはない。実際に、この数がｗと似たものになることを示すことができる。これらのパラメータに基づいて、複数の細分度を有するインデックス付けの追加の記憶域オーバーヘッドを計算することができる。図４の（ａ）に示すテーブルに関する計算は以下の通りである。

【００５５】

行数[4(a)]　　＝Ｄ　．．．（９）

全体サイズ[4(a)]　＝（１＋ｖ＋ｗ）Ｄ　．．．（１０）。

【００５６】即ち、サイズは図２の（ａ）に示すテーブルと同じである。一方、図４の（ｂ）に示すテーブルのサイズは、以下の通りである。

【００５７】

行数[4(b)]　　＝W／ｆ　．．．（１１）

全体サイズ[4(b)]　＝（１＋ｄｆ）・W／ｆ　．．．（１２）。

【００５８】辞書にあるワードのエントリの数は、ワードが意味概念に統合されるために、要素ｆに基づいて減少する。しかし、意味概念毎の文書の数は、その要素とほぼ同じ分だけ増加する。結果的に、このテーブルのサイズは、図２の（ｂ）に示すテーブルと同じようなものになる。より高いレベルの細分度においては、図４の（ａ）と（ｂ）に示すテーブルは、それぞれ図２の（ａ）と（ｂ）に示すテーブルであることに注意すべきである。最後に、図４の（ｃ）に示すテーブルの記憶域の見積りは、以下の式（１３）、（１４）のように計算される。

【００５９】

行数[3(b)]　＝Ｖ（Ｖ－１）／２＋Ｖ・（W／ｆ）＋（W（W－１）／２ｆ^2）　．．．（１３）

全体サイズ[3(b)]　＝（１＋２＋ｑ）・Ｖ（Ｖ－１）／２＋（１＋２＋ｑｆ）・Ｖ・（W／ｆ）＋（１＋２＋ｑｆ）（W（W－１）／２ｆ^2）　...（１４）

。

【００６０】基本的に、Ｓ－Ｓ型、Ｓ－Ｐ型、又はＰ－Ｓ型の共起用語の全ては、要素ｆに基づいて圧縮され、図３の（ｂ）に示すテーブルと較べて実質的に小さな容量になる。

【００６１】最終的に、本発明によれば、図３の（ｂ）に示すテーブル以外が必要とされる。一方、基本的なクエリー拡張技法は、図２及び図３に示すテーブルを全て必要とする。従って、本発明の方法を採用した場合の記憶域に関するコストは、図４の（ａ）と（ｂ）に示すテーブルの記憶域の増加分だけ大きくなるが、図４の（ｃ）に示すテーブルのサイズが小さくなるので、前記コストの増加分は部分的に埋め合わせられる。節約の正確な数値は、前述した種々のパラメータの値に依存する。最悪の場合でも、追加の記憶域は、基本的なクエリー拡張技法を使用した場合の記憶域の２倍よりかなり小さい。

【００６２】前述のインデックス付け技法は、ワードが単一の意味のみを有することを仮定して議論されている。しかし、ワードは、通常複数の意味を有している。例えば、「ｂａｎｋ」というワードは、金融機関（銀行）、または川岸として解釈される。複数の意味を有するワードについて考慮するため、意味的ワードリストのワード（図３で示されている）が、図４の（ａ）に示す複数の概念番号に属するものとする。例えば、「ｂａｎｋ」は、Ｓｅｍ１０とＳｅｍ２０に関連付けられるものとする。こうした複数の意味を考慮して、クエリーの拡張を実行するために、クエリーが、複数の異なる概念番号に属する１つのワードを含む場合、その異なる概念番号のそれぞれが、クエリーの処理の際に考慮されるべきである。

【００６３】上記説明においては、インデックス技法は、ＮＥＣのＯＯＤＢＭＳの上位層に実装されており、意味的ワードリスト内のワードが、ポインタによって概念番号に関連付けられている。また、冗長なデータは記憶されておらず、ポインタに関する記憶域のコストも非常に低いものである。ＷｏｒｄＮｅｔは、あるワードに

対する類義語を様々な意味の解釈で提供し、その意味が使用される頻度に応じてランク付けする。例えば、「ｂａｎｋ」は、川岸よりも金融機関として、より多く解釈される。最も一般的な意味解釈が、現在の実行に用いられる。しかし、データ構造を拡張することもできる。

【００６４】前で述べられた以外の、意味のグループ化を考慮に入れることができる。図４では、類義語によるクエリーの拡張だけが考慮されている。ＩＳＡ、ＩＳ＿ＰＡＲＴ＿ＯＦ等の、他の型の意味の緩和を考慮することもできる。図４の（ａ）に示す形態のテーブルを、様々な意味のグループ化（例えば、１つはＩＳＡに関して、１つはＩＳ＿ＰＡＲＴ＿ＯＦに関して）に関して複数生成することができる。また、１つのテーブルを様々な意味のグループ化に関して使用することもできる。類義語及び上位概念語の両方によってクエリーの拡張を行う場合、複数のテーブルに対してルックアップが行われる。

【００６５】ワード・ミスマッチの問題に対処するため、クエリー処理技法は、関連ワードを用いてクエリーのワードを拡張する必要がある。結果的に、元のクエリーのワードに対する関連性によって文書をランク付けする、追加タスクが実行されうる。次に、拡張されたクエリーの処理が、本発明による３つのタスク、即ち、クエリーの拡張、クエリーの処理、及び結果のランク付けとして提供される。

【００６６】まず、クエリーの拡張について説明する。図６は、従来のクエリー拡張技法の元でのクエリーの拡張の例を示している。「ｃａｒとｄｅａｌｅｒというワードを含む文書を検索する」というクエリーが修正され、ｃａｒとｄｅａｌｅｒに関連するワードが追加されている。意味的に類似する関連ワードと、構文的な共起関係を有する関連ワードは、図３に示すテーブルを用いて決定される。本発明による、複数の細分度を有するクエリー拡張技法の元でのクエリー拡張の例は、図７に示されている。複数の細分度を有するクエリーの拡張技法は、ｃａｒとｄｅａｌｅｒというワードを、図３の（ａ）に示すテーブルを用いて概念Ｓｅｍ１とＳｅｍ２に変換する。ワードを、そのワードに対応する、より上位レベルの意味概念に変換した後、図４の（ｃ）に示すテーブルを用いて、その意味概念が、構文的関係を含むように拡張され、元のクエリーにある固有名詞が、共起テーブルからの関連ワードを含むように拡張される。

【００６７】辞書にあるワードと辞書にないワードの両方を含むクエリーＱが与えられた場合、Ｑは、以下の式（１５）で表現される。

【００６８】

$$Q=(s_1\wedge...\wedge s_m)\wedge(p_1\wedge...\wedge p_n) \quad ...(15)$$

式（１５）では、s_iは辞書にあるワードを表し、p_jは辞書にないワードを表す。更に、クエリーＱには、辞書にあるワードがｍ個あり、辞書にないワードがｎ個ある。こうしたクエリーが与えられると、複数の細分度を有するクエリー拡張技法が、以下のように実行される。

【００６９】１．Ｑにある各s_i（$i=1,...,m$）を、図３の（ａ）に示すテーブルから得られた、そのs_iに対応する、より上位レベルの意味概念に置換する。このように置き換えられた概念のそれぞれをC_iと表記する。

【００７０】２．ステップ１で得られた各C_i（$i=1,...,m$）に対して構文的関連を有するワードを、図４の（ｃ）に示すテーブルを用いて求め、追加することによってＱを拡張する。Ｓ－Ｓ型のエントリは、概念の追加に寄与し、Ｓ－Ｐ型のエントリは、固有名詞に寄与する。

【００７１】３．各p_j（$j=1,...,n$）と共に生じる、構文的関連を有するワードを、図４の（ｃ）に示すテーブルを用いて求め、追加することによってＱを拡張する。Ｐ－Ｓ型のエントリは、概念の追加に寄与し、Ｐ－Ｐ型のエントリは、固有名詞に寄与する。

【００７２】４．冗長なクエリーのワード又は概念をＱから除去する。

【００７３】本発明によって拡張されたクエリーは、従来の技法によって拡張されたクエリーと較べて、よりコンパクトで、チェックすべき項目が少ない。それは、クエリーのワードが、より粗い細分度におけるエンティティに変換されているからである。結果的に、本発明により拡張されたクエリーのクエリー処理のコストは、一層小さいものになる。次に、従来の技術のクエリー拡張、及び本発明の、複数の細分度を有するクエリー拡張において導入されるエンティティ（ワード又は概念）の数が見積られる。前述のように、より上位レベルの意味概念の元でグループ化された、辞書にあるワードの平均数は、ｆで表される。ここで、ワードに意味的に関連付けられた、より上位レベルの概念の平均数をｇとし、ワードに関連付けられた、構文的関連を有する固有名詞の平均数をｈとする。そこで、基本的なクエリーの拡張（ＢＱ）の元でのＱにおけるワードの数は、以下の式（１６）に示すように、ステップ１、２、及び３で発生する拡張の合計である。

【００７４】

$$\text{ワード数}[BQ]=(mf)+m(g+h)+n(g+h) \quad ...(16)$$

ここで、第１項は、辞書にあるｍ個のワードのそれぞれが意味的に類似するｆ個のワードに置換されるために生じる。第２項は、辞書にあるｍ個のワードのそれぞれに対し、辞書にある共起ワード、及び辞書にない共起ワードが（ｇ＋ｈ）個追加されるために生じる。第３項は、（ｇ＋ｈ）個の共起ワードを追加したｎ個の固有名詞のそれぞれに対応する。同様に、複数の細分度を有するクエリー拡張（ＭＧＱ）の元でのＱにおけるワードと概念

の数は、以下の式（１７）で表される。

【００７５】

ワード数［ＭＧＱ］＝ｍ＋ｍ（ｇ／ｆ＋ｈ）＋ｎ（ｇ／ｆ＋ｈ）　．．．（１７）

ここでは、使用されている類似のワードの組に関して、より上位レベルの意味表現を用いているので、圧縮要素ｆが現れることが、実質的に大きく異なる点である。従って、複数の細分度を有するクエリー拡張技法によって、クエリーに含まれるワード／概念の数は、基本的なクエリー拡張技法によるものよりも厳密な意味で少なくなっている。図４の（ｃ）のテーブルにおいて、ワード毎の固有名詞の数が小さければ、本発明の技法によるクエリーの複雑さは、要素ｆに基づいて軽減される。

【００７６】今度は、クエリー処理について説明する。従来の厳密なマッチングに基づくクエリー処理では、クエリーに関連する検索の述語に関する条件を満たさないことが分かるとすぐに、検索処理を終了する。検索は類似性に基づいているので、実際のＩＲにおいては、そうではない。特にユーザは、ユーザの検索基準に、部分的にでもマッチした結果を見ようとするものである。従って、Ｎ個のワードを有するクエリーに対して、Ｎ回のルックアップが必要であり、これは、検索の述語におけるブール条件には依存しない。更に、部分的なマッチングが支援されているので、クエリー処理の後に、ランク付け処理を追加する必要がある。ランク付け技法は、文書の内のどのワードがクエリーにマッチするかということと、そのワードの文書内での頻度に関する情報を必要とする。

【００７７】ここで、２つの技法において、クエリーを処理する際のルックアップのコストについて分析する。２つの要因のために、処理コストには基本的な違いがある。この２つの要因は、以下に示すものである。

【００７８】・基本的なクエリー拡張におけるワード数が、複数の細分度を有するクエリー拡張におけるワード数より多いこと。

【００７９】・ルックアップが行われる、それぞれのテーブルのエントリ数が２つの技法で異なること。

【００８０】ここで、前述したクエリーＱのルックアップ・コストの見積りを行う。テーブルは、平衡探索構造で組織化されており、テーブルのルックアップ操作は、テーブルの行数に応じて対数的に変化するものと仮定する。従って、前述の見積り式を用いた、基本的なクエリー拡張においてＱを実行する際のルックアップ・コストは、以下の式（１８）、（１９）のようになる。

【００８１】

ルックアップ・コスト(Q,BQ) = mf log(行数[2(b)]+(m+n) (g+h) log(行数[3(b)]) ...（１８）

= mf log(W+V)+(m+n) (g+h) log(V(V-1)/2+VW+W(W-1)/2) ...（１９）。

【００８２】同様に、複数の細分度を有するクエリー拡張においてＱを実行する際のルックアップ・コストは、以下の式（２０）、（２１）のようになる。

【００８３】

ルックアップ・コスト(Q,MGQ) = m log(行数[4(b)]+(m+n) (g/f+h) log(行数[4(c)]) ...（２０）

= m log(W/f+V)+(m+n) (g/f+h) log(V(V-1)/2+V W/f+(W(W-1))/2f²) ...（２１）。

【００８４】辞書に有るワードのルックアップの回数が、ＭＧＱ内の要素ｆによって少なくなり、ルックアップの実行対象となる２つのテーブルのサイズが小さくなるので、ＭＧＱにおけるクエリー処理のコストがＢＱにおけるコストより小さくなるのは明らかである。

【００８５】次に、本発明のランク付け方法について説明する。クエリー処理段階において、より粗い細分度でのワードの表現が、関係のない文書を除去するのに用いられる。しかし、候補となる文書は、それらが２つの条件、即ち、より粗い細分度レベルにおいて「ｃａｒ」と「ｄｅａｌｅｒ」を含むという条件を満たすので、同じランクを有する。これは、クエリー処理の結果として好ましいものではない。従って、ランク付けの段階では、候補となる文書内にある元のワードがアクセスされ、それがランク付けに使われる。

【００８６】図８では、以下の条件を満たすキーワードを有する、４つの、候補となる文書が示されている。

【００８７】条件：（Ｓｅｍ１　∨　Ｆｏｒｄ　∨　Ｂｕｉｃｋ）∧（Ｓｅｍ２　∨　Ｆｏｒｄ　∨　ＢＵＩＣＫ）。

【００８８】最初のマッチング・キーワードがランク付けのために検索される。従って、（「ｃａｒ」、「ｄｅａｌｅｒ」）、（「ａｕｔｏ」、「ｄｅａｌｅｒ」）、（「ａｕｔｏ」、「ｓａｌｅｓ　ｏｆｆｉｃｅ」）、及び（「Ｆｏｒｄ」、「ｓｈｏｗｒｏｏｍ」）が、関連性の程度をランク付けするのに用いられる。

【００８９】候補となる文書は、クエリー内のワードを有する文書内においてマッチしたワードについての緩和の程度に基づいてランク付けされる。

【００９０】例えば、緩和の程度は、Ｅ＜Ｓｅ＜Ｓｙ＜Ｘ（即ち、厳密なマッチング＜意味的緩和＜構文的緩和＜マッチングなし）の順で定義される。ここで、クエリ

一のワードに関し、より高いレベルで緩和がされたワードを用いたクエリーの結果は、ユーザに対して、より関連のないものを含むことになる。しかし、緩和の程度の順と定義は、アプリケーションの要件によって任意である。候補となる文書を探すのに、より小さな緩和が用いられるほど、候補となる文書のランクは、より高くなる。図8の下部に、「ｃａｒ」と「ｄｅａｌｅｒ」というワードを有する文書に最も高いランクが与えられている。これは、候補のワードがクエリーのワードに厳密にマッチしたからである。「ａｕｔｏ」と「ｄｅａｌｅｒ」というワードを有する文書は２番目に高いランクが与えられている。これは、クエリーのワード「ｃａｒ」とマッチさせるため、１つのワードのみに、意味的な緩和（即ち、クエリーの用語を、意味的に関連する用語と入れ替える）が必要とされるからである。他のランク付けに関しては、図８に示すように行われる。

【００９１】ランク付け技法は、以下の２つの基準に基づいて行われる。

【００９２】・与えられたクエリーＱのキーワードに関して、Ｑと文書Ｄｏｃ１にあるキーワードＷｏｒｄ１、Ｄｏｃ２にあるＷｏｒｄ２、Ｄｏｃ３にあるＷｏｒｄ３、Ｄｏｃ４にあるＷｏｒｄ４との間の関係がそれぞれ、厳密なマッチング、意味的なクエリー緩和によるマッチング、構文的なクエリー緩和によるマッチング、及びマッチングなしである場合、文書は、Ｄｏｃ１＞Ｄｏｃ２＞Ｄｏｃ３＞Ｄｏｃ４の順にランク付けされる。

【００９３】・Ｍ個の文書、Doc_i（ｉ＝１，．．．，Ｍ）と、文書Doc_iにそれぞれ対応する、クエリーにマッチするキーワード数、$Match_i$（ｉ＝１，．．．，Ｍ）に関するランク付け（スコア）は、$Match_1 > Match_2 > Match_3 ... Match_{M-1} > Match_M$である場合、$Doc_1 > Doc_2 > Doc_3 ... Doc_{M-1} > Doc_{M0}$となる。

【００９４】２つのキーワードを備えたクエリーを使用する、前述したランク付け技法に基づけば、図９に示すような、２つのワードを有するクエリーで文書を検索する場合の２次元ランク付けグラフが生成される。クエリーの拡張をしないと、スロット（Ｅ，Ｅ）内の文書だけが検索される。クエリーの意味的拡張と構文的拡張の両方を用いると、文書がスロット（Ｘ，Ｘ）にない限り、関連する全ての文書が検索される。

【００９５】このランク付けグラフは、行列として表されている。Ｎ個の用語を有するクエリーに関して、ランク付けグラフは、Ｎ×４の行列、Ｍ（ｉ，ｊ）（ｉ＝０．．．Ｎ、ｊ＝０．．．３）によって表される。例えば、図９のランク付けグラフは、行列Ｍ（ｉ，ｊ）（ｉ＝０．．．２、ｊ＝０．．．３）として表されている。例えば、スロット（Ｅ，Ｅ）、（Ｓｅ，Ｅ）、（Ｓｅ，Ｓｙ）、及び（Ｘ，Ｘ）は、行列内で、それぞれスロット（３，３）、（２，３）、（２，１）、及び（０，０）として表されている。この表現によれば、各文書は以下のように簡単にランク付けできる。

【００９６】・スロット（ｎ，ｍ）内の文書に対して、ｍが０から３の間である場合、これらの文書のランクは、スロット（ｉ，ｊ）（ｉ＝０．．．ｎ、ｊ＝０．．．３）内の文書より高いスコアになる。

【００９７】・スロット（ｎ１，ｍ１）内の文書のランクのスコアは、ｎ１≧ｎ２かつｍ１≧ｍ２である場合、スロット（ｎ２，ｍ２）内の文書のスコア以上になる。

【００９８】このランク付けグラフの表現は、市販の視覚化ツールによって実現される。例えば、Ｃｏｎｅ　Ｔｒｅｅｓと呼ばれる視覚化方法は、３次元のランク付け表現に関する奥行きを追加することによって変更されうる。詳細については、１９９３年４月、Communications of the ACM, Vol. 36, No. 4, ページ57-71の、G. G. Robertson他による「Information Visualization Using 3D Interactive Animation」を参照のこと。

【００９９】このランク付け技法に基づけば、図９の上部のスロット内の結果は、下部における結果よりも高いスコアでランク付けされる。しかし、図９において同じクラスに属するスロットの結果をランク付けするのは困難である。図１０は、そのようなランク付けがどのように行われるかを示している。結果的に示されたスロットは、更にクラスに分類され、そこで、同じクラスのスロットが同じランクを有するようにされる。

【０１００】本発明によるクエリー処理は、図１０に示すクラス構造を用いて、クラス毎に連続して行われる。ユーザが２つのキーワードを持つクエリーを発行し、上位５０個の結果が検索されるように要求した場合について考える。図１０を参照すると、クエリー・プロセッサは最初に、クラス０に検索結果を生成する可能性がある。検索結果が５０より多い場合、クエリー・プロセッサは、クエリー拡張タスクを実行することなく処理を終了することができる。クラス０における検索結果の数が５０に満たない場合、クエリー・プロセッサはクラス１（例えば、スロット（２，３）及び（３，２））にその結果を生成することができる。検索結果（例えば、クラス０及びクラス１における）の総数が５０より多い場合、クエリー・プロセッサは、更にクエリー処理をすることなく、処理を終了する。クエリー・プロセッサは、スロット（２，３）及び（３，２）内の結果も連続的に生成することができることに注意すべきである。つまり、クエリー・プロセッサは、スロット（２，３）の結果を最初に生成することができる。検索結果の総数が５０を越える場合、クエリー・プロセッサは、スロット（３，２）内に結果を生成することなく、処理を終了することができる。クエリー・プロセッサは、検索結果の総数が５０を越えるまで、又は最後のクラスに達するまで、前述のように、残りのスロット及びクラスから、更なる結果の生成を続けることができる。

【０１０１】上記の例が、１つのキーワードが他のキーワードより重要であるとして、ユーザによって変更される場合、クエリー・プロセッサが検索結果のスロットを検索する順序は、その変更に応じて修正される。例えば、ユーザが、キーワード１はキーワード２より重要であると指定した場合、クラス内の水平的なクエリー処理の順序は、図１１に示すように導出される。即ち、この例では、クエリー・プロセッサが、最初にスロット（３，２）に検索結果を生成する。次に、検索結果の総数が５０に満たない場合、クエリー・プロセッサはその後、スロット（２，３）に結果を生成する。

【０１０２】図１２は、本発明が実行されるシステムの物理的構成を示している。こうしたシステムは、文書の集合体を記憶するデータベース１２０６を含んでいる。このデータベースは、概念（例えば、意味的又は構文的概念）及び、文書の集合体に対するそれらの関係を記憶するためのインデックス１２０８を含んでいる。システムは更に、インデックス１２０８を生成し、より上位の細分度を有する概念と文書の集合体に対するそれらの関係を含むインデックス１２０８を生成するためのインデクサ１２１０を含む。プロセッサ１２０４は、ユーザ・インタフェース１２０２を介してユーザから指定されたクエリーを受信するのに使用される。プロセッサ１２０４は、次に、クエリーを処理し、ランク付け機能を実行する。クエリーの結果とランク付け機能は、ユーザ・インタフェース１２０２を介して再びユーザに表示される。

【０１０３】当業者は、本発明の実施が、図１２で例示された実施例に限られるものではないことを理解することができる。実際、当業者は、本発明の範囲から逸脱することなく、他の代替ハードウエア環境を使用して同様の効果を得ることができる。例えば、上述の、様々な機能が別個の要素によって実行され（例えば、クエリー処理とランク付け機能が、別の構成要素で行われる）、又は単一の要素によって行われる（例えば、単一のプロセッサが、インデックス付け、クエリー処理、及びランク付け機能を実行する）。

【０１０４】要するに、本発明は、入力された文書に関するキーワードの組の元の有効性（適合率、及び再現率）、用語の意味を含む辞書、及びクエリーを保持したまま、効果的に複数の細分度に亘るインデックス付け（インデックス領域の節約）と、クエリー処理（処理時間の節約）を用いてクエリーの拡張を支援するための新しい技法を提供する。

【０１０５】本発明による複数の細分度に亘るインデックス付け技法とクエリー処理技法によって、クエリーが単純化されるため、ワードの関連を示すインデックスのサイズがより小さくなり、クエリーの処理時間が短くなる。また、本発明のランク付け技法が、文書内に最初からあるワードに基づくため、ランク付けの結果に一貫性が保たれる。

【０１０６】ここまでの開示及び教示から、当業者が、本発明に対して様々な、他の変更及び修正をすることができることは明らかである。従って、本明細書では本発明のいくつかの実施例についてのみ述べているが、本発明の意図及び範囲を逸脱することなく、本発明に対して様々な変更を考えることができる。

【０１０７】

【発明の効果】本発明によれば、ワード・ミスマッチの問題と、結果的に生じるクエリー処理の非効率さを解決するために、小さなサイズのインデックスを使用して、効率的なクエリー拡張が行われる。具体的には、クエリー内に指定されたワードと意味的に類似し、構文的に関連のあるワードを用いて、そのクエリーを、物理的ではなく概念的に拡張し、結果的に、関連する文書を逃すことを少なくすることができる。

【図面の簡単な説明】

【図１】情報検索に関するワード・ミスマッチの問題を示す図である。

【図２】厳密なマッチングの情報検索システムで、従来より使用されているインデックスの例を示す図である。

【図３】従来の情報検索システムで使用するために、ワードを意味的に類似した概念、及び構文的に関連する拡張にグループ化することによって得られたインデックスの例を示す図である。

【図４】本発明において、より効率的にクエリー処理を行うために必要なインデックス構造を示す図である。

【図５】共起ワードのインデックスのエントリをマージする処理を示す図である。

【図６】従来の情報検索システムにおけるクエリー拡張処理を示す図である。

【図７】本発明による、複数の細分度を有するクエリー拡張技法を用いたクエリー拡張処理を示す図である。

【図８】本発明による、ランク付け処理を示す図である。

【図９】２つのワードを有するクエリーのランク付けを表す２次元グラフである。

【図１０】連続的なクエリー処理の順序を示す図である。

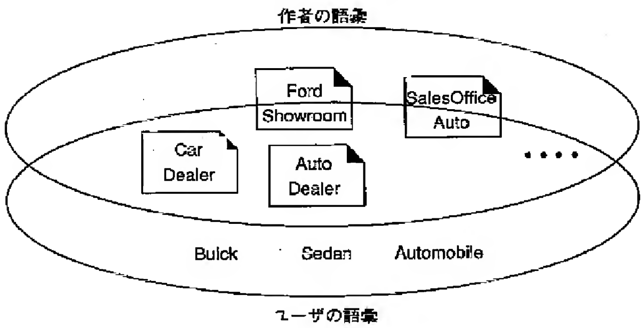
【図１１】キーワードがあるレベルの重要度に割り当てられている場合の、連続的なクエリー処理の順序を示す図である。

【図１２】本発明が実施可能な一実施形態の物理的構成を示す図である。

【符号の説明】

１２０２　ユーザ・インタフェース
１２０４　プロセッサ
１２０６　データベース
１２０８　インデックス
１２１０　インデクサ

【図1】



【図2】

(a)

文書番号	ワードリスト
Doc1	Ford,Showroom
Doc2	Auto,SalesOffice
Doc3	Car,Dealer
Doc4	Auto,Dealer
:	:

(b)

ワード番号	文書リスト
Ford	Doc1
Showroom	Doc1
Auto	Doc2,Doc4
Dealer	Doc3,Doc4
:	:

【図3】

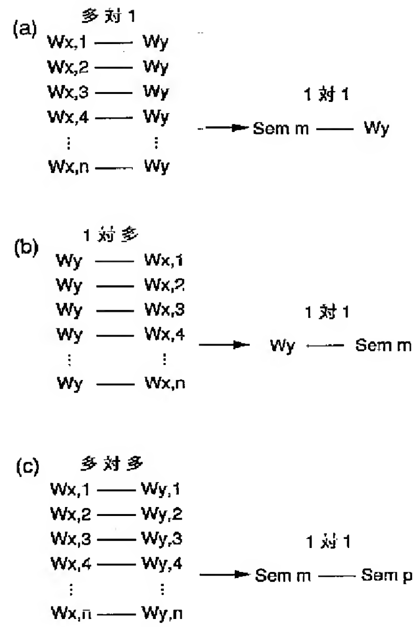
(a)

概念番号	意味的ワードリスト
Sem1	Car,Auto,Automobile,Sedan
Sem2	Dealer,Showroom,SalesOffice
Sem3	Garage,Parking
:	:

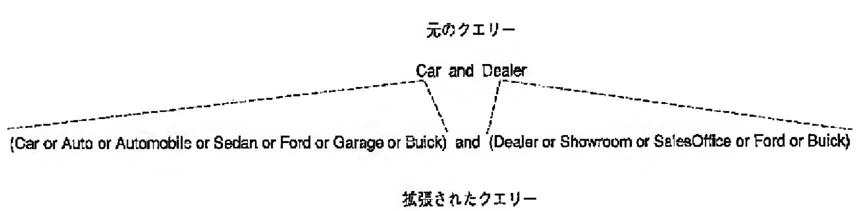
(b)

構文番号	構文的ワードリスト
Syn1	Buick,Car
Syn2	Car,Garage
Syn3	Auto,Garage
Syn4	Ford,Car
Syn5	Ford,Auto
:	:

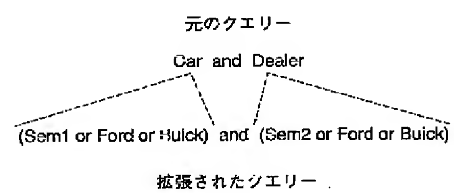
【図5】



【図6】



【図7】



【図4】

(a)

文書番号	概念・固有名称リスト
Doc1	Ford,Sem2
Doc2	Sem1,Sem2
Doc3	Sem1,Sem2
Doc4	Sem1,Sem2
:	:

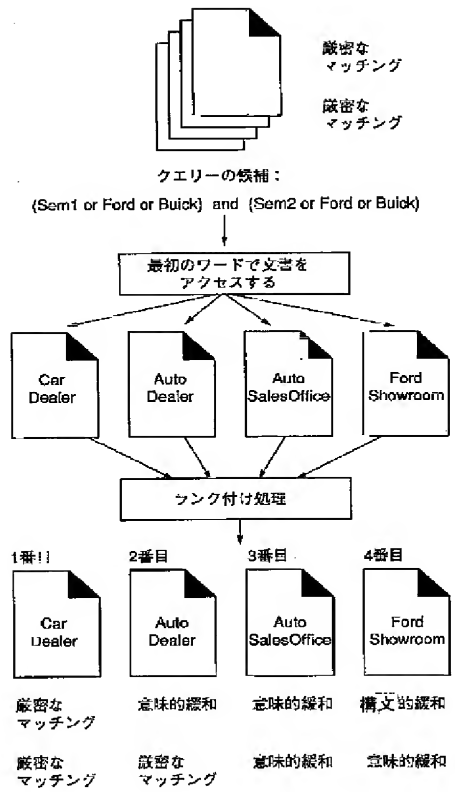
(b)

概念番号	意味別文書リスト
Sem1	Doc2,Doc3,Doc4
Sem2	Doc1,Doc2,Doc3,Doc4
:	:

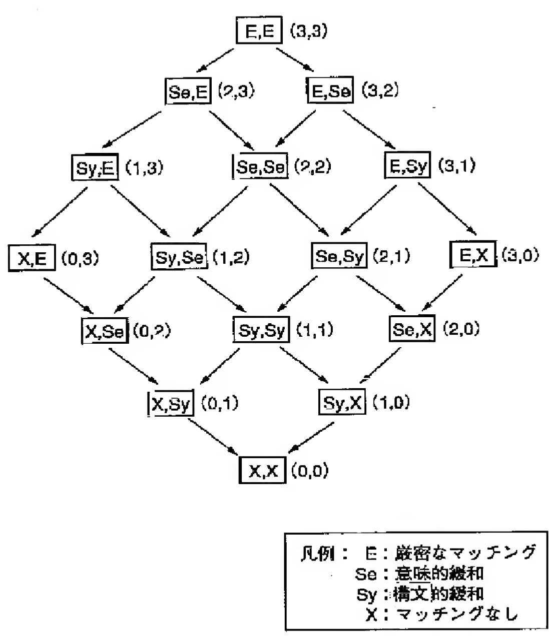
(c)

共起番号	構文別ワード・リスト
Syn1'	Buick,Sem1
Syn2'	Sem1,Sem3
Syn3'	Ford,Sem1
Syn4'	Sem7,Sem21
Syn5'	Ford,Buick
:	:

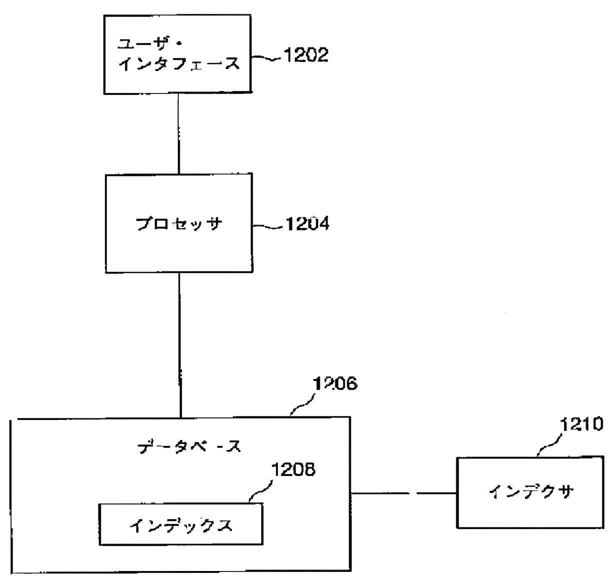
【図8】



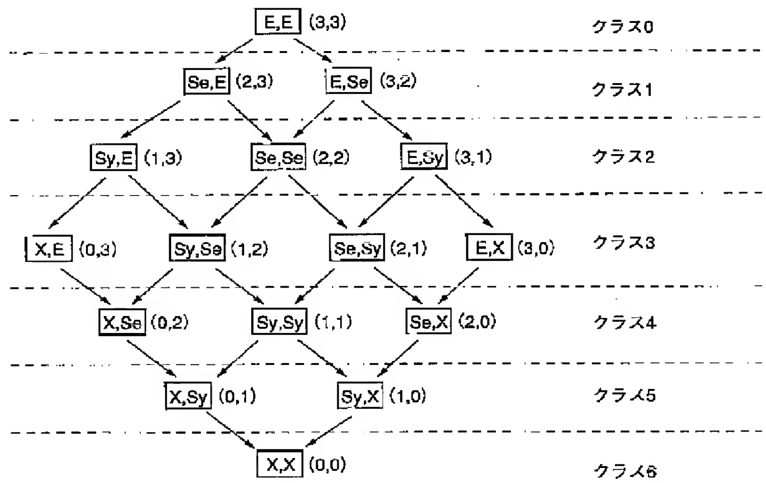
【図9】



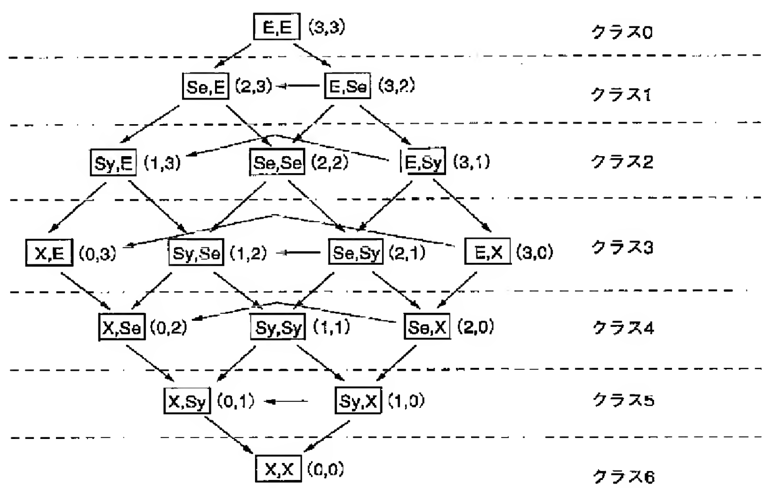
【図12】



【図10】


E,E (3,3)
クラス0
Se,E (2,3)
E,Se (3,2)
クラス1
Sy,E (1,3)
Se,Se (2,2)
E,Sy (3,1)
クラス2
X,E (0,3)
Sy,Se (1,2)
Se,Sy (2,1)
E,X (3,0)
クラス3
X,Se (0,2)
Sy,Sy (1,1)
Se,X (2,0)
クラス4
X,Sy (0,1)
Sy,X (1,0)
クラス5
X,X (0,0)
クラス6

【図11】


E,E (3,3)
クラス0
Se,E (2,3)
E,Se (3,2)
クラス1
Sy,E (1,3)
Se,Se (2,2)
E,Sy (3,1)
クラス2
X,E (0,3)
Sy,Se (1,2)
Se,Sy (2,1)
E,X (3,0)
クラス3
X,Se (0,2)
Sy,Sy (1,1)
Se,X (2,0)
クラス4
X,Sy (0,1)
Sy,X (1,0)
クラス5
X,X (0,0)
クラス6